ISTRUZIONI, NOTE E SEGRETI TECNICI DEI PIÙ DIFFUSI DRIVE PER COMPUTER

COMMODORE 64-128

DRIVE

1541/1571

I LISTATI DI TUTTE LE ROUTINE PER USARE AL MASSIMO DELLA POTENZA I DRIVE COMMODORE

L. 5.000

SOMMARIO

INTRODUZIONE

Al momento dell'acquisto di questo libro, immagino abbiate pensato di riuscire in poco tempo ad impadronirvi di ogni tecnica teorica e pratica relativa al lettore di dischi nato per il Commodore 64. Purtroppo devo deludervi: anche se vi sentirete molto piu' capaci ed in grado di lavorare disinvoltamente, capirete che la "sapienza" vi verra' quasi esclusivamente dalla pratica, ed in fondo questo libro vuole essere un ottimo aiuto per scoprire cose che col tempo, forse un po' troppo, avreste finito per scoprire anche voi da soli.
Le pubblicazioni finora uscite sull'argomento, riservate agli addetti ai lavori o contenenti delle cose talmente ovvie da risultare inutili, non hanno assolutamente esaurito il filone letterario delle novita' che sono scopribili sia sul Commodore 64 che sul 1541.
Ecco perche' mi sono deciso a scrivere questo libro: non ho mai studiato informatica sui banchi di scuola, possiedo il 64 da quasi tre anni, e mi ritengo poco piu' di un appassionato del settore: perche' non aiutare coloro che si avvicinano a questa interessante materia usando un linguaggio facilmente abbordabile, scaturito dall'esperienza pratica e non da accademismi teorici?
Tutto quello che conosco mi e' stato dettato dall'esperienza e da qualche (pochissimi) indispensabile testo, quale puo' essere ad esempio una mappa di memoria, o il DOS del 1541.
In questo libro verranno trattati un'infinita' di argomenti, forse troppi per un solo libro: la gestione dell'informazione, i metodi di protezione dei programmi, gli errori e i test su di essi, la manipolazione diretta del disco, il caricamento ed il salvataggio rapido dei programmi (la modifica del DOS), i copiatori, i test meccanici sul drive, ed altri argomenti ancora.

Claudio Cerroni

Capitolo 1: LA MEMORIA SU DISCO

Il lettore di dischi presenta notevoli ed apprezzabili differenze rispetto al registratore: come memoria di massa, infatti, e' molto piu' rapido e flessibile, lavorando con i files non in modo sequenziale ma in una maniera completamente diversa, per cui e' possibile caricare una parte di dati che ci interessano senza caricare o comunque passare in rassegna l'intero contenuto memorizzato sul supporto magnetico; per usare una similitudine che possa meglio illustrarvi la differenza, provate un po' a pensare alla musica registrata su nastro e su disco: su nastro saremo costretti a cercare sequenzialmente la canzone desiderata facendo scorrere, se e' il caso, l'intera cassetta; su disco bastera' appoggiare la puntina del back-up sulla giusta traccia per ascoltare cio' che desideriamo.

In un drive, e cosi' nel 1541, e' il DOS (Disk Operating System) che pensa ad indirizzare, dietro nostra istruzione, la testina di lettura verso l'informazione da leggere (di questo ne parleremo in modo piu' approfondito nel Capitolo dedicato ai comandi del DOS 2.6).

Un'altra caratteristica che fa del 1541 una macchina piu' utile del 1520 (il registratore dedicato al 64) e' la velocita' di caricamento e di gestione dei dati. Infatti, con il registratore per alcuni programmi occorrono anche 18-20 minuti per il completo caricamento di un programma , mentre col lettore di dischi sono sufficienti 2 minuti e mezzo, o al massimo 3 minuti per occupare l'intera memoria disponibile nel C-64. Ma, sempre in questo libro, troverete delle utility che vi daranno la possibilita' di attendere ancora meno, tramite la riprogrammazione del DOS residente.

L'interno del 1541 non e' un segreto per nessuno: basta infatti aprirlo per vedere come lavora e come e' fatto fin nei minimi particolari. L'unico problema consiste nell'automatica invalidita' della garanzia (le viti sono coperte o segnate dalla ceralacca, e sono testimoni dell'avvenuta apertura dell'apparecchio), ed in ogni caso vi spieghero' in modo semplice, come si compone internamente il 1541.

E' sottinteso che se riscontrerete anomalie di funzionamento (potrete testare il funzionamento grazie ad una serie di listati sempre presenti in questo libro) vi consiglio di non operare direttamente sulla macchina ma di rivolgervi al vostro rivenditore o riparatore di fiducia, anche se i tempi saranno probabilmente lunghi.

Gli schemi meccanici e componentistici non sono pubblicabili: per questo vi daro' una descrizione sommaria per spiegarvi in modo semplice come lavori il 1541.

Il disco, una volta inserito nella fessura e dopo che avrete chiuso lo sportellino, verra' premuto da un tampone di feltro contro il supporto inferiore, entrando in contatto con la testina di lettura. Quest'ultima si muove lungo un braccio orizzontalmente, e si trova al di sotto del disco; quindi osservando un qualsiasi dischetto, sappiate che la faccia di lettura e' quella inferiore, ed e' principalmente quella che dovrete assolutamente salvaguardare

dalla polvere, dal sudore delle dita, ed altro.

Molte persone hanno l'abitudine di usufruire dell'altra faccia del dischetto, intagliando simmetricamente una fessura simile per dimensioni a quella gia' presente; non tutti pero' conoscono gli inconvenienti a cui e' possibile andare incontro.

Prima di tutto, il dischetto ha una resa fisica inferiore, perche' costruito per girare in un solo senso; inoltre la garza presente al suo interno, che funziona come tergi-polvere, viene rivoltata ogni volta che il disco gira. Ci si puo' accorgere anche dal diverso rumore prodotto dal dischetto; sappiate che girando nel giusto verso la parte magnetica si pulisce, nel verso contrario, invece, si sporca nuovamente. Ecco perche', come affidabilita', un supporto tagliato dura meno.

Come possiamo pulire le testine di lettura?
Esistono in commercio molti kit completi pratici e convenienti, consistenti in un disco di garza imbevuto di alcool da far girare all'interno dell'apparecchio per una trentina di secondi. Dato che questa operazione sarebbe da ripetere frequentemente, quasi non conviene smontare l'apparecchio per operare manualmente, anche se la pulizia risulterebbe senz'altro migliore.

Ricordatevi, in ogni caso, se intendete aprire il drive, di non toccare nulla con le dita, ne' con oggetti metallici, per non pregiudicare la funzionalita' delle parti megnetiche (vedremo piu' avanti cosa fare esattamente).
Se desiderate eliminare i possibili rumori che, specialmente a motorino caldo, caratterizzano il lavoro del 1541, potete farlo lubrificando con prodotti al silicone le cinghie di trasmissione. Ricordatevi che il drive e' una macchina molto delicata, quindi abbiate l'accortenza di non scuoterla, lasciarla cadere o cose simili, perche' e' abbastanza facile che si guasti.

Il supporto magnetico

Il dischetto, anche se non sembra, al suo interno non e' affatto rigido: viene costruito con un materiale molto simile al nastro magnetico, assumendo una forma circolare per possedere quelle caratteristiche che prima abbiamo riportato. Esistono in commercio tantissime marche diverse di supporti magnetici: si diversificano tra loro non tanto per la qualita' della parte magnetica, ma piuttosto per la qualita' della meccanica, concernente la qualita' del cartoncino che sostiene il resto, la resistenza alle variazioni climatiche (ricordiamoci che una delle peggiori caratteristiche del nostro 1541 e' rappresentata dalla pessima dispersione del calore), la presenza di un anello rinforzatore al centro, il tipo di garza interna e la lubrificazione necessaria ad un perfetto scorimento del disco...

Ho eseguito, a questo proposito, una serie di test su alcuni dschetti in commercio, senza avere la pretesa di essere infallibile: dobbiamo pensare che alcuni supporti differiscono anche dalla densita' della parte magnetica, direttamente proporzionale al loro costo.

MARCA DISCO	MOD.	QUAL.MAGN	QUAL.CART.
MEMOREX	1D-40	BUONA	BUONA
SCOTCH 3M	744D-0	OTTIMA	BUONA
DYSAN		OTTIMA	OTTIMA
FLEXETTE	4281	BUONA	DISCRETA
NASHUA	MD1D	OTTIMA	DISCRETA
G.M.C.		DISCRETA	DISCRETA
DATALIFE		DISCRETA	OTTIMA
SOTTOMARCHE	----	--------	--------

E' abbastanza facile testare la qualita' magnetica di un dischetto: battendo il listato che segue avrete l'opportunita' di controllare l'effettiva affidabilita' di un supporto.

Il programma non fa altro che scrivere e leggere sul nostro disco effettuando una serie di passaggi blocco per blocco verificando se l'informazione ne risulta alterata o addirittura distrutta. Il nostro 1541 non e' un drive di grandi pretese: il 99% dei supporti va benissimo, e l'unico giudice puo' essere il tempo.

Ad essere fortunati, una buona marca puo' durare dai tre ai cinque anni, mentre delle sottomarche a malapena arrivano a sei mesi; l'ideale sarebbe conservare due copie per ogni supporto, una per l'uso piu' o meno quotidiano, l'altra come archivio, da usare il meno possibile; ma anche in questo modo non e' detto che l'informazione nel tempo non si degradi.

-------------------------- (vedi listato 1) --------------------------

Al momento della formattazione (vedi prossimo capitolo), operazione indispensabile per poter usare un dischetto, la superficie magnetica viene divisa dal DOS del 1541 in 35 tracce concentriche, divise ognuna in un certo numero di settori, numerio che varia a seconda della posizione piu' o meno interna della traccia, che puo' essere percio' piu' o meno estesa.

Le traccie dalla numero 1 alla 17 posseggono 21 settori ognuna (possiamo anche chiamarli blocchi), quelle dalla 18 alla 24 ne possiedono 19, quelle invece dalla 25 alla 30 ne possiedono 18, ed infine le traccie dalla 31 alla 35 ne hanno 17, per un totale complessivo di 683 blocchi, di cui disponibili 664, poiche' la traccia 18 e' riservata alla Directory ed alla BAM.

Ogni settore o blocco e' composto da 256 bytes, ossia cellette di memoria, per cui su ogni dischetto sono disponibili 169.984 bytes.

La fessura circolare presente nel cartoncino permette di riconoscere se il dischetto e' soft-sectored o hard-sectored: soft-sectored presenta un solo forellino, usato dall'unita' a dischi per avere un inizio, da cui creare via software una serie di traccie e di blocchi; hard-sectored, invece, hanno una serie di forellini che rappresentano gia' la divisione del dischetto in traccie predeterminate via hardware.

Il 1541 usa esclusivamente dischi soft-sectored, in teoria; personalmente ho provato diverse volte ad usare l'altro tipo di supporto senza avere alcun inconveniente.

Il contenuto del disco, come accennatosopra, e' memorizzato sulla diciottesima traccia, e per caricare la lista e' necessario battere:

```
LOAD"$",8
```

Dando successivamente il comando LIST apparira' sullo schermo l'intera sequenza dei programmi, preceduti dalla loro occupazione in blocchi e seguiti dalla sigla del tipo di file. Il numero 8 caratterizza il tipo di periferica utilizzata (1 per il registratore, da 8 a 11 per il drive, 3 o 4 per la stampante).

Sempre sulla traccia 18, nel blocco Ø, risiede la BAM (Block Availability Map) che contiene informazioni sulla disponibilita' di memoria sul dischetto. Nel capitolo dedicato all'editor, ed alla protezione e sprotezione dei programmi su questo tipo di supporto, analizzeremo a fondo l'importanza di questa traccia, e come sia facilmente modificabile fino a plasmarla ai nostri desideri.

I comandi diretti del 1541

Vediamo un po' come agire sul 1541 tramite il C=64: innanzi tutto occore aprire un canale di comunicazione tra il computer e la periferica: il drive infatti, cosi' come la stampante e a differenza del registratore, e' una periferica quasi autonoma; possiede infatti una propria memoria interna che diventa operativa al momento dell'accensione.

Il canale per impartire ordini al 1541 si apre cosi':

```
OPEN 15,8,15
```

dove il numero 8 e' il device number, il primo 15 e' il canale di comando principale ed l'altro e' quello secondario.

I buffers disponibili sono 5, dove due vengono usati come controllo. Per gli indirizzi secondari, 16 in tutto, sono utilizzabili tranne lo Ø, l'1 e il 15, che apre un canale di comando speciale, usato per conoscere lo status del dischetto, e per altre cose che vedremo in seguito.

Come gia' abbiamo accennato, un dischetto prima di essere usato va inizializzato, va cioe' suddiviso dal 1541 in traccie e blocchi (o settori), con la creazione della BAM e la sua memorizzazione sulla traccia 18 (bl.Ø).

Per inizializzare un supporto, dobbiamo naturalmente aprire un canale di scrittura, ed utilizzare il comando "NEW:nome del disco,numero dell'ID", dove il nome del disco non puo' superare i 16 caratteri, e il numero dell'ID (identificatore del supporto), formato da due cifre o lettere, impedisce che il DOS confonda le BAM di due diversi dischi.

La formattazione dura esattamente 90 secondi, ed e' possibile ascoltare gli scatti meccanici che caratterizzano il passaggio della testina da un settore all'altro. Se un disco ha gia' subito questo processo e l'operatore ha intenzione di riutilizzare il supporto cancellando tutta la memoria, e' possibile, omettendo il numero di identificazione, riformattare senza attendere di nuovo il minuto e mezzo necessario.

Oltre tutto, la formattazione cancella in modo IRREVERSIBILE l'informazione memorizzata, mentre la riformattazione azzera semplicemente la BAM, ricostruibile, in caso d'errore di operazione, come vedremo in seguito grazie ad una serie di comandi al DOS del 1541.

Un altro comando e' l'INIZIALIZE, che non inizializza il disco (abbiamo appena visto che e' NEW il comando da usare), ma riporta la testina sulla traccia 18, la traccia centrale del nostro dischetto. E' un comando che non si usa molto spesso, e che non comporta assolutamente la scrittura sul disco: semplicemente viene memorizzato dal DOS il nome del disco e il numero di ID, leggendoli appunto dalla diciottesima traccia.

Occorre ricordare che ogni volta che apriamo un canale, quando questo non ci serve piu' occorre chiuderlo con il comando:

```
CLOSE numero canale
```

e questo per evitare messaggi di errore specialmente durante l'esecuzione di un programma.

Per riordinare i blocchi ed i settori di un disco, dobbiamo dare una VALIDATE, tramite il comando:

```
PRINT#15,"V"
```

e sul nostro supporto tutti i files rimasti aperti per errore verranno cancellati e la BAM aggiornata.

IMPORTANTE: non usate questo comando se avete dei file RANDOM sul disco! I file RANDOM, distribuiti blocco per blocco tramite un'allocazione predefinita, verrebbero de-allocati, e perdereste l'informazione contenuta.

Per duplicare i nostri files, o concatenarli fra loro, dovremo invece usare il comando COPY, con una sintassi diversa a seconda di quello che desideriamo.

Per duplicare, digiteremo:

```
PRINT#15,"COPY:NUOVO NOME=VECCHIO NOME"
```

Per concatenare invece:

```
PRINT#15,"C:NUOVO N.=1.VECCHIO N.,2.VECCHIO NOME"
```

Per correggere i nomi gia' dati ai programmi, ci sara' utile il RENAME, nella seguente forma:

```
PRINT#15,"R0:NUOVO NOME=VECCHIO NOME"
```

Per cancellare invece i files indesiderati, useremo il comando SCRATCH cosi':

```
PRINT#15,"S0:NOME DEL FILE"
```

A scanso di brutte sorprese, sara' opportuno far seguire un comando VALIDATE, in modo tale da riorganizzare efficacemente la BAM del disco.

Se poi siete cosi' fortunati da possedere due (o piu') drive, per farli lavorare collegati occorrera' modificare tramite via software o via hardware il numero di periferica (DEVICE). Farlo via software e' facilissimo, basta seguire queste istruzioni:

1) Spegnete le periferiche lasciando accesa quella di cui volete modificare il numero di accesso

2) Aprite un canale di comunicazione con:

```
OPEN 15,8,15
```

3) Digitate il seguente comando (che vi sara' meglio spiegato nel prossimo capitolo):

```
PRINT#15,"M-W"CHR$$(119)CHR$(0)CHR$(2)CHR$(9+32)  CHR$(9+64)
```

4) Non spegnete piu' la periferica; accendete invece la successiva e modificate il 9 con il numero desiderato (8, 10 o 11)

Capitolo 2: I comandi del DOS 2.6

Ricapitolando quanto abbiamo detto nel precedente capitolo, ecco una piccola tabella che vi aiutera' efficacemente nel vostro lavoro di programmazione.

```
COMANDO-FORMA-COSA PROVOCA

NEW - PRINT#1,"N0:nome,ID" - formattazione del
                             disco vergine.
NEW - PRINT#1,"N0:nome"    - riformatta un disco
                             non vergine.
INIZIALISE - PRINT#1,"I    - memorizza BAM e ID
                             in traccia 18
VALIDATE - PRINT#1,"V      - riorganizza la BAM e
                             chiude i files aperti
RENAME - PRINT#1,"R:nuovo nome=vecchio nome -
                             cambia nome al file
COPY - PRINT#1,"C:nuovo nome=vecchio nome - copia
                             un file
COPY - PRINT#1,"C:nuovo f.=1o file,2o file-collega
                             2 files tra loro
SCRATCH - PRINT#1,"S:nome file - cancella il file
                             dalla directory
```

A questo proposito, vi sara' molto utile il programma qui di seguito proposto; riassume tutti i comandi che abbiamo fino ad ora incontrato, e vi permettera' di prendere un po' piu' di confidenza con il vostro drive, senza paura di sbagliare.

-------------------- listato n. 2 - utility disk --------------------

Nel digitare il listato, incontrerete tutti i comandi diretti che fino ad ora hanno occupato un posto rilevante nella nostra conversazione.
Dovrete riconoscere quanto sia facile usarli senza per questo sentirsi imbarazzati; bastera' comunque un po' di pratica per rendervi tutto piu' pratico, e perche' no, divertente.

I comandi di servizio

BLOCK-READ

Questo comando va a leggere un blocco in uno dei buffer della memoria; per la coretta esecuzione, occorre aprire il canale di comando, ed un canale ad accesso diretto; le istruzioni sono le seguenti:

```
100 OPEN 15,8,15
110 OPEN 2,8,4,"#"
120 INPUT "TRACCIA"; T
130 INPUT "SETTORE"; S
140 PRINT#15, "B-R:"4;0;T;S
```

dove quest'ultima riga trasferisce un blocco di dati al buffer.

```
150 GET#2, C$
160 IF ST=0 THEN PRINT C$;:GOTO 150
170 CLOSE 2: CLOSE 15
```

Tutti i parametri occorrenti a qest'istruzione devono essere separati dal punto e virgola. Nel momento in cui il blocco e' stato trasferito in un buffer, usiamo l'istruzione INPUT# o GET# per estrarli.

E' preferibile quest'ultima quando si e' incerti sulla quantita' di dati da caricare, perche' e' molto facile superare il massimo degli 80 bytes che caratterizza le possibilita' di gestione dell'informazione da parte dell'istruzione INPUT# (GET#, invece, legge un byte alla volta).

BLOCK-ALLOCATE

La BAM, ogni volta che viene salvato un file sul disco, viene modificata e salvata in trccia 18; questo comando ha lo specifico compito di indicare se nella BAM esiste un blocco che sta per essere usato, e che contiene gia' delle informazioni.

In poche parole, usiamo l'istruzione BLOCK-ALLOCATE per controllare, prima di scrivere sul blocco di una trccia, se in quel blocco esistono gia' dei dati, che andrebbero persi. In caso questo blocco risulti impegnato dalla BAM, verra' visualizzato un messaggio contenente il sucessivo blocco disponibile.
Ma ecco una piccola routine:

```
100 OPEN 15,8,15
110 INPUT"TRACCIA :";TR
120 INPUT"SETTORE :";SE
130 PRINT#15, "B-A:"0;TR;SE
140 INPUT#15,E,EM,EM$,T,S
150 PRINT E,EM$,T,S
160 CLOSE 15
170 END
```

Il messaggio sara' un "OK" in caso il blocco risulti disponibile.

BLOCK-WRITE

Seguendo la logica di programmazione, eccoci ad osservare il comando di scrittura su disco, che e' sempre meglio controllare PRIMA di scrivere se il blocco e' disponibile con una BLOCK-ALLOCATE.
Aggiungiamo quindi alla routine precedente le seguenti linee di programma:

```
150 IF EM$="OK" THEN 180
160 TR=T : SE=S
170 IF TR=0 AND SE=0 THEN PRINT "DISCO PIENO"
    : GOTO 250
180 PRINT"TRACCIA :";TR,"SETTORE :";SE
190 OPEN 2,8,4,"#"
200 INPUT C$
210 IF C$="*" THEN 240: REM FINE INSER. DATI
220 PRINT#2,C$
230 GOTO 200
240 PRINT#15, "B-W:"4;0;TR;SE
250 CLOSE 2:CLOSE 15
```

BLOCK-EXECUTE

Questo comando carica un blocco del supporto magnetico contenente una routine in linguaggio macchina, facendola eseguire fino a trovare un'istruzione RTS.

```
PRINT#15,"B-E:"15;8;1;4
```

BUFFER-POINTER

Se ci occorre leggere un byte di un blocco che non sia il primo, invece di creare un LOOP di lettura per arrivare a quel byte, possiamo "puntare" tramite questo comando, direttamente al byte desiderato:

```
PRINT#15, "B-P:"ind.sec.;n.byte
```

Ad esempio, per puntare al 18o byte di un blocco, digiteremo:

```
PRINT#15,"B-P:"4;18
```

BLOCKS-FREE

Questo comando rende disponibile qualsiasi blocco del dischetto, chiedendo alla BAM di contrassegnarlo come disponibile. E' chiaro che e' un comando abbastanza pericoloso da usare, se non si ha un po' di dimestichezza con la gestione dell'informazione su disco.
La forma e' la seguente:

```
100 OPEN 15,8,15
110 INPUT"TRACCIA :";T
120 INPUT"SETTORE :";S
130 PRINT#15, "B-F:"0;T;S
140 CLOSE 15
150 END
```

Ecco un listato che vi permettera' di pulire il disco dai files indesiderati, senza rischiare di rovinare la rimanente informazione.

------------------ listato n. 3 - pulizia disco ---------------------

I Comandi alla Memoria-Disco

Come e' chiaramente visibile dalla foto, il drive 1541 dispone di un micro-processore 6502, simile al 6510 del C=64, che ha una RAM di 2K e un DOS contenuto su due ROM.

Abbiamo gia' visto come parte di questa memoria venga utilizzata internamente, per il controllo dela BAM o le operazioni sui files.
Un'altra perte di questa memoria e' riservata ad operazioni particolari: dobbiamo tener presente che se vogliamo scrivere un programma in linguaggio macchina in quelle aree di memoria, abbiamo a disposizione cinque pagine di memoria, di 256 bytes ciascuna.
E' sicuramente piu' affidabile utilizzare i primi 4 buffer di memoria, che nel caso non abbiate gia' aperto canali di memorizzazione, sono disponibili e non creano problemi al DOS (cosa che puo' accadere nel quinto buffer).

I comandi che tratteremo di seguito sono di esclusivo utilizzo da parte di persone che hanno gia' avuto a che fare con lo studio di DOS o che abbiano una certa dimestichezza con il linguaggio macchina.

MEMORY-WRITE

Corrisponde al comando POKE del normale BASIC, con la differenza che MEMORY-WRITE puo' trasferire nella memoria del 1541 fino a 344 caratteri, tutti codificati tramite il Codice ASCII.

Ha delle limitazioni (come tutti questi comandi) perche' e' solamente un'estensione del linguaggio macchina. Per questo motivo dovremo indicare le locazioni di indirizzo tramite il codice esadecimale, digitando il byte basso e quello alto.

```
OPEN 15,8,15
PRINT#15, "M-W"CHR$(00)CHR$(03)CHR$(1)CHR$(250)
CLOSE 15
                         ↑↑       ↑↑            ↑
                        byte     byte         valore
                       basso     alto       desiderato
```

In questo modo caricheremo il valore 250 nella locazione di memoria n.768 decimale, corrispondente a $0300 in esadecimale.

MEMORY-READ

Questo comando e' l'equivalente del comando PEEK: legge infatti una determinata locazione di memoria del 1541.

La sintassi e' simile all'istruzione precedente:

```
100 OPEN 15,8,15
110 PRINT#15, "M-R:"CHR$(00)CHR$(03)
120 GET# 15,C$
130 PRINTC$
140 CLOSE 15
```

MEMORY-EXECUTE

Viene usato per caricare e fare eseguire un programma in linguaggio macchina da parte del 1541; deve terminare con un RTS, che restituisce il comando alla memoria del C-64.
La sintassi e' identica a quella appena vista:

```
100 OPEN 15,8,15
110 PRINT#15, "M-E:"CHR$(00)CHR$(03)
120 CLOSE 15
```

Un'ultima raccomandazione: queste ultime tre istruzioni vanno digitate SOLO nella forma abbreviata, altrimenti otterremmo un messaggio di errore da parte del DOS del 1541.

I comandi dell'utente

Questi comandi riassumono tutti quelli appena visti e commentati, e presentano delle caratteristiche che li fanno preferire ai programmatori per la maggior versatilita':

U1 = BLOCK-READ

U1 legge tutte le informazioni del blocco, compresi i due byte iniziali che puntano a quello successivo (cosa che B-R non faceva). E' possibile in questo modo concatenare facilmente la lettura dei blocchi contenenti una medesima informazione.

U2 = BLOCK-WRITE

U2 ha la medesima sintassi di B-W, ma permette di scrivere una serie di dati senza chiudere il blocco. In questo modo e' molto piu' facile correggere dati gia' presenti sul disco, evitando un lavoro di ricerca tra i blocchi successivi interessati.

U3 - U9 e U

Rappresentano la funzione di M-E, saltando ad una specifica locazione iniziando l'esecuzione del programma (in l.m.).

------------------- listato n. 4 - ordin.directory -----------------

Ecco un listato che vi permettera' di ordinare alfabeticamente le vostre directory. Quasi tutti i comandi sinora visti vengono impiegati, in modo tale da facilitarvi il compito di cpire il funzionamento pratico di quanto abbiamo visto fino ad ora solo in teoria.

RICAPITOLANDO

COMANDO	SINTASSI
BLOCK-READ	PRINT#15,"B-R:"is;dr;t;s
BLOCK-ALLOCATE	PRINT#15,"B-A:"dr;t;s
BLOCK-WRITE	PRINT#15,"B-W:"is;dr;t;s
BLOCK-EXECUTE	PRINT#15,"B-E:"is;dr;t;s
BUFFER-POINTER	PRINT#15,"B-P:"is;n.byte
BLOCK-FREE	PRINT#15,"B-F:"dr;t;s
MEMORY-WRITE	PRINT#15,"M-W"CHR$(b.b.)CHR$(b.a.) CHR$(n.b.)C HR$(dati)
MEMORY-READ	PRINT#15,"M-R"CHR$(b.b.)CHR$(b.a.)
MEMORY-EXECUTE	PRINT#15,"M-E"CHR$(b.b.)CHR$(b.a.)

e i comandi dell'utente:

U1 = BLOCK-READ
U2 = BLOCK-WRITE
U3 = Locazione $0500
U4 = Locazione $0503
U5 = Locazione $0506
U6 = Locazione $0509
U7 = Locazione $050C
U8 = Locazione $050F
U9 = Locazione $FFFA
U = Routine accensione

Capitolo 3: I Files su disco

Il file caratterizza il supporto magnetico: infatti, nel momento in cui vogliamo salvare un programma memorizzato nel nostro computer, trasformiamo questo programma in un file, in una particolare sequenza di dati che il computer traduce n instruzioni.

Vi sono 4 tipi di files: i files PROGRAMMA, i CANCELLATI (DELete), i SEQUENZIALI, quelli riservati al commento della directory (USeR), e i RELATIVI, non accessibili dal Basic V2.0 del C=64.

Ognuno di questi files e' caratterizzato da un numero esadecimale memorizzato sulla disectory, e permettono al DOS di riconoscerli durante la ricerca; questo numero puo' essere facilmente modificato per nascondere o far riapparire un file a seconda delle nostre esigenze.
Ma questo e' un argomento che approfondiremo nel capitolo dedicato all'Editor per il 1541.

I listati che seguono vi permetteranno di leggere ogni tipo di file diverso dal PRG (che e' leggibile tramite il semplice caricamento). Particolarmente utile e' il file di lettura per sequenziali: infatti e' possibile sfruttarlo in una routine di lettura continua da disco, per esempio nella programmazione di un'avventura. In questi casi la famntasia non ha limiti.

---------------------- listato n. 5 - legge seq.---------------------

---------------------- listato n. 6 - legge rel.---------------------

---------------------- listato n. 7. - legge D.B.---------------------

---------------------- listato n. 8 - legge hdr ---------------------

I files in BASIC e in Assembly

Occorre distinguere i programmi in due categorie: quelli che sono composti da istruzioni BASIC, e quelli invece che sono strutturati in linguaggio macchina.

Affinche' un programma in BASIC possa partire, digiteremo:

```
LOAD"nome del prg.",8

RUN
```

dove 8 e' il numero della periferica.

Il programma verra' automaticamente memorizzato a partire dalla locazione esadecimale $0801, tipica dei programmi in BASIC. I problemi sorgono quando il file e' in linguaggio macchina; una caratteristica di questi files, infatti, e' proprio la partenza in punti della memoria disparati, variabili dalla locazione $03FD fino alla fine della memoria disponibile.

Prima di tutto, il comando di caricamento sara':

LOAD"nome del prg.",8,1

dove il numero 1 consente la memorizzazione a partire dalla locazione originale del programma.

Battere il comando RUN darebbe un errore di sintassi, in quanto non c'e' niente alle locazioni di partenza del BASIC. Dobbiamo per forza di cose conoscere la locazione di partenza del prg. in linguaggio macchina, per poi digitare:

SYS locaz.iniz.decimale

Nella memoria del C=64 c'e' una porzione di memoria dedicata al linguaggio macchina, che il piu' delle volte viene utilizzata come sede di routine di ordinamento o di ricerca dati: in esadecimale, a partire da $C000 in poi, in decimale dalla locazione 49152. Di conseguenza, se un programma in l.m. e' memorizzato in quella zona, bastera' digitare da tastiera o da programma:

SYS 49152

Ma torniamo ai vari tipi di files: sono riconducibili a tre.

Il primo, file programma, di cui abbiammo gia' parlato, ha la sintassi di apertura e di chiusura automatica, poiche' e' il DOS del 1541 che compie tutte le procedure necessarie.

Il secondo, file sequenziale, utilizzabile solo per la memorizzazioni di dati, ha gia' bisogno di una serie di comandi di apertura e di chiusura, e dipende sempre da un programma.

Il terzo, anch'esso usato per la gestione di dati, file ad accesso casuale (RANDOM), ha la caratteristica di non occupare la memoria del computer, permettendo una gestione analitica dell'informazione, riducendo anche i tempi d'attesa nell'elaborazione; purtroppo ha una serie di caratteristiche istruzioni che se non usate al meglio, rischiano di invalidare il nostro lavoro di digitazione.

Come sopra accennato, esistono altri due tipi di files, i relativi e quelli dedicati al commento dell'utente (USeR), ma non ci interessano, almeno per il momento.

I files sequenziali

L'istruzione da usare e' la seguente:

OPEN 2,8,4,"0:NOME FILE,SEQ,W"

per aprire un file in scrittura, mentre useremo:

OPEN 2,8,4,"0:NOME FILE,SEQ,R"

per aprirlo in lettura ("W" e' l'abbreviazione di WRITE, scrivere, e "R" l'abbreviazione di READ, leggere). La sintassi dell'apertura dei canali gia' la conosciamo, e non ci dilungheremo di nuovo.

Come per i file programma, ci puo' capitare di voler riscrivere o rimpiazzare un file sequenziale gia' presente sul dischetto;

basterà' che aggiungiamo il simbolo "@" al nome del file facendogli precedere il numero zero.

Nel caso il file non esista sul disco, il DOS aprira' normalmente il file ignorando il simbolo aggiuntivo.

Una caratteristica spesso presente nei programmi gestionali in commercio e' rappresentata dall'uso di stringhe per la creazione di files. Occorrera' pero' ricordarsi che per concatenare il comando di apertura di canale con la stringa tramite il simbolo "+". Il comando OPEN, infatti, deve essere un'unica stringa.

La gestione di un file da parte del 1541 avviene nel modo seguente: esistono 5 buffer disponibili, come abbiamo gia ' visto, ed ognuno di questi buffer rappresenta un canale di passaggio per l'informazione. Quest'ultima, infatti, viene dapprima trasferita nel buffer, fino al suo riempimento, e poi dal buffer al disco; l'operazione viene completata dalla chiusura del file, che deve avvenire puntualmente, per evitare brutte sorprese.

I file ad accesso casuale (RANDOM)

Utilizzando il comando OPEN, e' possibile utilizzare un dischetto memorizzando i dati blocco per blocco, trasferendo l'informazione dai buffer disponibili (4 su 8) al supporto magnetico.

Per evitare un errore di sistema, e' sconsigliabile aprire piu' di 4 canali per volta.

Abbiamo gia' visto come si utilizzano i comandi del DOS 2.6 per organizzare la scrittura e la lettura di dati; il listato che segue non e' altro che la pratica applicazione di quanto finora appreso.

---------------------- listato n. 9 - archivio random ---------------

Capitolo 4: Gli errori...

Per gli appassionati del C=64 che possiedono gia' da qualche tempo il drive, questo capitolo ha l'aria di essere uno dei piu'interessanti.

Parleremo di errori: "Errare humanum est" diceva un celebre luogo comune romano, e gli errori hanno infatti sempre accompagnato l'attivita' umana. A causa di sbagli, inavvertenze, chieramente non volute, e' possibile, anche nel campo informatico, commettere errori imperdonabili, capaci di compromettere un lungo lavoro di archiviazione, o di programmazione. Cancellare per sbaglio un file da un disco e' capitato a tutti, ma non tutti sono in grado di far "resuscitare" il file modificando l'informazione direttamente sul dischetto.

------------------ listato n.10 - recupera files -------------------

Questo listato vi permettera', senza tecniche abbastanza complicate (che affronteremo nelcapi.tolo dedicato all'Editor) di recuperare i file cancellati dal vostro disco-dati, o di controllare ildisco prestatovi da un amico desideroso di nascondervi qualcosa... L'importante e' che in questo caso voi non scriviate nulla sul disco, in modo tale da non comprometterne il contenuto. In pratica, quando diamo il comando SCRATCH di un file, il DOS non fa altro che cancellare i primi due bytes di riconoscimento del file, rendendo disponibili i blocchi da lui occupati. Con l'Editor saremo in grado di riscrivere quei due bytes e di riportare alla luce della directory il programma cancellato.

Per il momento, pero', utilizzeremo questo programma, che tra le altre cose ha il pregio di non sbagliare ed evita un ulteriore peggioramento dello stato del disco, specialmente a chi non e' avvezzo a queste cose.

Gli errori su disco

Dopo aver parlato degli errori umani, eccoci a parlare di quelli computerizzati: ebbene, sembra impossibile, ma anche il nostro drive puo' sbagliare, e gli errori vengono trascritti fedelmente sul supporto magnetico!

Il DOS del 1541 possiede una serie dimessaggi di errore che avvertono l'operatore dell'eventuale presenza di anomalie, sul disco o nella memoria del drive.

Nel manuale che accompagna il lettore di dischi, troviamo la tabella completa, che ci interessa solo in parte, poiche' tratteremo solo di alcuni di questi messaggi.

Uno dei problemi che da sempre hanno assillato i programmatori, e' stato l'abitudine abbastanza screanzata di alcuni appassionati di copiare il software altrui, magari frutto di giorni o mesi di duro lavoro. Questi programmatori hanno pensato bene di proteggere in qualche modo il proprio lavoro, sfruttando le caratteristiche del DOS del 1541.

L'errore su disco, letto in un certo punto del programma, rappresenta una delle protezioni piu' usata.

Un disco formattato dal 1541 contiene, abbiamo detto, 683 blocchi, dove ogni blocco, ed ogni settore, nel suo header ha una serie di bytes che lo rendono riconoscibile e classificabile nel momento della lettura o della scrittura.

Quando noi salviamo dei dati su un supporto, e in seguito siamo in grado di rileggerli, possiamo dire di essere di fronte ad un dischetto efficiente, che il DOS non ha problemi nella sua gestione. Nel caso qualcosa non funzionasse (ad es. il supporto viene a contatto con una superficie metallica, o si sporca di polvere, o viene unta la parte magnetica dal sudore delle nostre dita), invece di leggere l'informazione che era presente, il DOS ci visualizza un messaggio d'errore, essendo impossibilitato ad eseguire il nostro ordine.

Molti programmatori hanno pensato bene di sfruttare questa caratteristica: un errore viene provocato sul disco, chiaramente dove non c'e' informazione, ed il programma va a testarlo, prima di eseguire un certo passaggio; se l'errore non c'e' il programma non funziona correttamente.

Per compiere questa operazione, non occorre nulla di speciale; tramite delle routine software e' possibile simulare una serie di errori che possono essere comodamente testati in qualsiasi punto del programma.

Fino a tutto il 1984, sembrava che nessun operatore fosse in grado di duplicare un dischetto protetto in tal modo, ma in seguito ci si e' accorti di quanto fosse facile riprodurre gli errori con poche istruzioni in linguaggio macchina.

Ma come si comporta il drive quando incontra un messaggio simile?

Nel momento in cui la testina si posiziona su un blocco con l'header contenente un errore, il led luminoso comincia a pulsare, mentre la testina va a battere contro il fermo che stabilisce l'inizio della traccia 1 (e' lo stesso rumore che si sente all'inizio della formattazione). E' inutile dire quanto sia dannosa una sollecitazione del genere, soprattutto pensando che vi sono programmi che ad ogni passaggio testano un errore e rovinano inutilmente l'allineamento della testina.

Il DOS emette un messaggio di errore che letto da un'appropriata routine stabilisce se il codice e' quello giusto o no. In caso non vi sia un errore (la testina non fara' rumore posizionandosi sul blocco prescelto) il programma e' in grado di bloccarsi, di rovinare l'informazione sul disco, di mandare al diavolo l'operatore incauto resettando il sistema.

Fino a poco tempo fa era considerato un metodo efficace (e per molti lo e' ancora), ma in effetti proteggeva solamente il programma dall'esserecopiato, e non dall'essere sprotetto.

Ma vediamo quali sono gli errori, ed i messaggi che il DOS invia al computer quando li incontra.

I messaggi d'errore

Consultando la Guida per l'utente che accompagna il nostro 1541, ecco la nostra tabella con tutti gli errori segnalabili dal DOS 2.6.

ERRORE	SIGNIFICATO
0	- Tutto a posto, non vi sono errori.
1	- File cancellato (dopo uno SCRATCH)
2/19	- Numeri non utilizzati dal DOS.
20	- Manca l'intestazione del blocco (header).
21	- Errore di sincronismo (formattazione errata)
22	- Assenza di dati dal blocco.
23	- Errore di controllo dei dati.
24	- Errore di codifica del byte.
25	- Errore di scrittura verificato.
26	- Scrittura con protezione inserita.
27	- Errore di controllo nell'intestazione.
28	- Estensione dei dati nel blocco successivo.
29	- Non c'e' coincidenza nell'ID.
30	- Errore di sintassi di un comando.
31	- Invalidita' di comando.
32	- Lunghezza eccessiva linea di comando.
33	- Invalidita' nome di un file.
34	- Manca il nome del file.
39	- File non trvato.
50	- Record non presente.
51	- Overflow su di un record.
52	- File troppo grande.
60	- File aperto in scrittura.
61	- File non aperto.
63	- File gia' esistente.
64	- Manca verifica per il tipo di file.
65	- Blocco non disponibile.
66	- Traccia o settore non permesso.
67	- Traccia o settore non permesso dal sistema.
70	- Non vi sono canali disponibili.
72	- Directory o disco completo (pieno).
73	- DOS incompatibile in scrittura.
74	- Unita' non pronta (8050).

Gli errori che realmente ci interessano, perche' simulabili sulla superficie magnetica di un dischetto, sono solo sei, ed ora li esamineremo uno ad uno per capire come sono e come possono verificarsi.

20 : READ ERROR

Manca l'intestazione del blocco: il controllore cerca il blocco dei dati, ma non lo trova, per un errato indirizzo del settore o per la distruzione del blocco dei dati.

21 : READ ERROR

Errore di sincronismo, dovuto ad un'errata formattazione del dischetto. E' ottenibile da un drive disallineato, che non formatta completamente il supporto, lasciando senza sincronismo o la traccia 1 o la traccia 35 (le traccie estreme del disco). Si puo' ottenere anche aprendo lo sportellino del drive durante la formattazione di un disco.

22 : READ ERROR

Assenza di dati dal blocco, o errore di dati in scrittura. Viene comandato al controllore di andare in un settore e traccia sbagliati.

23 : READ ERROR

Errore di controllo nella lettura dei dati: indica che i dati hanno subito una deformazione (anche per cause fisiche esterne). Puo' essere causato anche da un'alimentazione non perfettamente isolata.

27 : READ ERROR

Errore di controllo nell'intestazione. Il blocco dati richiesto presenta un errore nel suo header, che non puo' essere letto dalla memoria del DOS.

29 : DISK ID MISMATCH

Anche questo e' un errore di formattazione, o di cattiva inizializzazione. Si verifica soprattutto quando inseriamo un dischetto non inizializzato (il DOS non riconosce l'ID) e diamo un comando di lettura.

-------------------- listato n. 11- trova errori --------------------

Questo piccolo programma e'in grado di analizzare blocco per blocco tutto il disco per segnalarci quanti e quali errori vi siano presenti.

Utilissimo per testare i dischi di dati, permette in un certo modo, di controllare se il nostro drive ha o meno difficolta' di lettura, specie nelle traccie esterne. I comandi usati sono gia' stati commentati, e li conosciamo gia'; sara' bene prendere una certa dimestichezza con essi perche' saranno i piu' usati nella gestione del controllo da parte dell'operatore, dei dati presenti sul dischetto.

----------------- listato n. 12- creatore di errori -----------------

Capiremo piu' a fondo l'utilita' di questo listato nel corso del prossimo capitolo. Per ora ci limiteremo a digitarlo, badando bene di non provarlo su un disco con dei files interessanti.

Sembrerebbe un paradosso: il DOS ci manda dei messaggi per avvertirci che sul disco vi sono degli errori, e noi ne siamo contenti!!!

Capitolo 5: Le protezioni

Nel capitolo precedente abbiamo accennato all'uso degli errori d'informazione come una protezione del software. Ma cos'e' una protezione ?

Fino ad oggi abbiamo assistito ad una gara incredibile tra protettori e sprotettori: ad un ritmo formidabile sono nate generazioni nuove di programmi e nuove tecniche di sprotezione. E' impossibile avere la soluzione finale di tutto cio': pobabilmente l'ultima risposta per quanto riguarda la duplicazione di supporti magnetici verra' da un "copiatore stupido", che non avra' la pretesa di interpretare l'informazione, ma si limitera' a copiare magneticamente il dischetto.

Ma neanche questa dovrebbe essere l'ultima soluzione: vedremo, infatti, come sia possibile modificare il DOS per impedire una corretta lettura dell'informazione, tramite la creazione di mezze traccie e di codifiche particolari dei bytes su disco.

Universalmente, e' possibile stabilire una classifica di livelli di protezione, e questo per aiutarci a capire le nostre possibilita' attuali.

1o LIVELLO

Programma completamente sprotetto, che puo' essere salvato tramite il LOAD ed un successivo SAVE, contenente solo Basic, o linguaggio macchina codificato in Basic. Sebbene questo livello possa considerarsi "non protetto", occorre tener conto di alcune cose: se il programma e' in Basic, e' possibile che si concateni con altri files, o che utilizzi file sequenziali o archivi di dati che non sono riproducibili tramite i due comandi sopra citati. Inoltre, il comando SAVE e' valido solo per il Basic, e non tiene conto di un'eventuale parte del programma in Assembler.

In ogni caso, con questo tipo di programma, un qualsiasi copy-files ha successo, anche se il modo piu' sicuro sarebbe un riproduttore di dati regolato dalla BAM.

2o LIVELLO

Fornito di un autostart e di un comando disabilitante il tasto RUN STOP. Non e' possibile caricarlo e salvarlo direttamente, ma occorre, usando un monitor, caricarlo in una locazione differente e riprodurlo. L'utente non e' in grado di bloccare il programma durante il suo funzionamento, e per vedere come e' strutturato il programma occorre, con un po' di fantasia, introdurre dati errati, o richiedere informazioni da periferiche staccate in modo tale che l'errore provocato possa riportare il controllo all'operatore. Solitamente un programma di questo genere prevede molte situazioni di errore, e lo scopo ultimo e' quello di non modificare il listato, e non quello di essere copiato.

La copia e' facilmente eseguibile da un copy-files, ed eventuali modifiche sono apportabili tramite un monitor (che ad esempio cancelli la parte iniziale del programma che provoca la partenza

spontanea del programma, nell'area di memoria antistante la locazione $0801).

3o LIVELLO

Dischetto contenente errori in alcuni blocchi, che impediscono a copiatori normali di proseguire la copia.

In pratica e' una protezione creata per bloccare la copia completa del disco (indispensabile con file CASUALI o RELATIVI), ed ha successo con i vecchi copiatori che non hanno una routine di controllo: incontrando un errore, infatti, cercano in tutti i modi di leggere ugualmente un'informazione che in realta' non esiste. I copiatori piu' evoluti, invece, saltano i blocchi "cattivi" senza leggerli.

Solitamente non ci sono di routines di controllo sul tipo di errore incontrato.

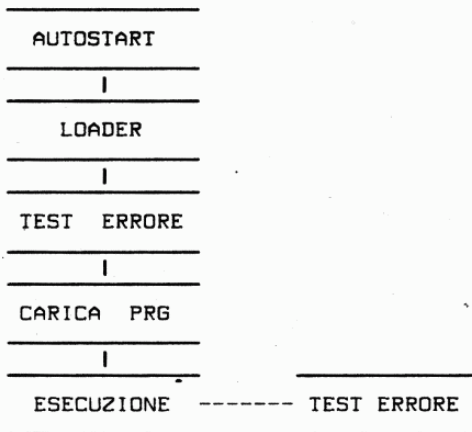
4o LIVELLO

Protezione simile alla precedente; per maggior sicurezza, viene testato il tipo di errore nel blocco e traccia desiderati.

E' il livello piu' diffuso tra il software oggi in circolazione. I programmi possiedono un autostart, non si possono fermare, ed in piu' controllano che ci sia un errore giusto al posto giusto.

I messaggi d'errore sono stati elencati nel precedente capitolo, e quelli sfruttabili per questo tipo di protezione sono l'errore 20, il 21, il 22, il 23, il 27 ed il 29. Per riprodurre dischetti di questo tipo si agisce in due modi: riproducendo fedelmente l'errore o gli errori (insieme all'informazione, oppure aggirando la routine (solitamente in linguaggio macchina, ma noi la tratteremo in Basic) che controlla l'esistenza dell'errore.

Un sistema molto usato consiste nel caricare un "loader" che prima di immagazzinare i dati del programma testa l'errore.

```
 __________________
   AUTOSTART
 __________________
        |
 __________________
     LOADER
 __________________
        |
 __________________
  TEST   ERRORE
 __________________
        |
 __________________
  CARICA   PRG
 __________________
        |
 __________________        __________________
   ESECUZIONE   -------  TEST ERRORE
 __________________        __________________
```

Se gli errori, o meglio i messaggi d'errore non sono quelli giusti, il loader puo' non caricare il programma, distruggere l'informazione sul dischetto, resettare il sistema , ecc. ecc.
In piu', il loader puo' decidere quali siano i blocchi da caricare ed in quale ordine, risultando la BAM cancellata. Il programma verrebbe cosi' caricato, mentre il loader controllerebbe se gli errori sono presenti, e tutto contemporaneamente.

In pratica si avrebbe una serie di settori da caricare ed una da testare, ottenendo un alto livello di protezione da copiature.

Per utilizzare questo tipo di protezione occorrono due programmi: il primo, un Editor, che analizzi la BAM e ci dica quali siano i blocchi contenenti l'informazione e quali invece siano disponibili per l'inserimento di errori; un secondo, un creatore di errori, che ci permetta di creare nel settore voluto il tipo di errore desiderato. Per finire, cosa piu' importante di tutte, una routine che, inserita all'interno del nostro programma (in Basic) ci permetta di controllare l'esistenza dei giusti errori sul dischetto.

Per copiare, invece, questo tipo di programmi, occorre utilizzare un lettore di errori, un riproduttore di supporti che non si fermi all'errore, un creatore di errori. Oppure, un sistema un po' piu' complicato, caricare tramite un monitor in linguaggio macchina i files, trovare la routine che testa il tipo di errore ed eliminarla o modificarla (ma questo tipo di tecnica e' riservata a quei pochi di voi che si intenderanno a fondo di questo linguaggio); ne parleremo piu' avanti, trattando l'uso dell'Assembler nella programmazione del 1541.

5o LIVELLO

Livello di protezione "avanzata" tramite l'uso di errori. Viene alterata quella che e' la normale procedura di formattazione del disco, producendo traccie extra (alcuni arrivano fino alla traccia 44!!!), speciali codifiche dei dati sul formato delle traccie, scrittura di dati sotto degli apparenti errori, la creazione, addirittura, di "mezze traccie" non leggibili normalmente dal DOS del 1541. Con questo tipo di protezione, pero', possono sorgere non pochi problemi di compatibilita', che rischiano di rendere inutilizzabile l'informazione contenuta dal supporto.

Sembrerebbe un'estensione del precedente livello, ma in effetti questo tipo di lavoro presuppone una profonda conoscenza del linguaggio macchina e del DOS, livello al quale sicuramente non e' possibile arrivare da parte dei semplici appassionati ma che, nonostante tutto, possono essere copiati utilizzando dei copiatori gia' presenti sul mercato (come abbiamo detto prima, e' una gara dove non ci sono regole o colpi proibiti, tra produttori e "copiatori").

Questo tipo di programma ha lo svantaggio notevole, come abbiamo gia' detto, di non risultare sempre leggibile su ogni drive di produzione Commodore; occorre anche pensare che i primi drive avevano la spiacevole propensione al disallineamento della testina, incontrando in lettura degli errori (parleremo piu' avanti di disallineamento e su come agire per limitarlo).

Per copiare questo tipo di programma occorre comportarsi come al livello precedente, usando pero' delle particolari routines che vanno al di la' dello scopo di questa pubblicazione. L'ideale, come sempre,

consisterebbe nel disassemblaggio del programma e nella sua correzione, ma bisogna esserne capaci.

6o LIVELLO

Programmi che richiedono, per il loro funzionamento, inserimenti di moduli hardware, solitamente tramite l'uso di pin di joystick o di cartuccie contenenti chip con memoria residente. Richiedono un sintetizzatore di formato, una "chiave" che permetta l'accesso dell'operatore al programma.

In questo modo non si utilizzano piu' gli errori e le routine di lettura, ma delle chiavi "hardware" presenti nelle porte esterne del C=64. Una protezione valida e che non penalizza assolutamente la velocita' di esecuzione del programma: testare un errore richiede un certo tempo (ed una serie di disturbi...), mentre testare la presenza di una chiave nella porta del joystick non dura piu' di qualche microsecondo, ed inserito in un programma gestionale e' praticamente irrilevante.

Parlando della presa per i joystick, ci sono 32 poli diversi, che possono essere circuitati, e corrispondono a 32 diverse protezioni. Per utilizzare questo tipo di protezione occorre conoscere un po' la meccanica degli spinotti, ed utilizzare la lettura della locazione 56320 per la porta n.1, e 56321 per la porta n.2.

Questa piccola routine vi indichera' come agire sul joystick per utilizzare questo ingegnoso dispositivo:

```
100 PRINT PEEK(56320): REM CONTROL PORT N.1
110 FOR A=0 TO 1000 : NEXT A
120 REM ↑ CICLO DI RITARDO PER LA LETTURA
130 GOTO 100
```

Per la copiatura di questi programmi occorre smontare la chiave hardware ed analizzarne la meccanica, tenendo pero' presente che e' possibile l'uso di schemi piu' complicati, con la programmazione di ROM sulla spina, per rendere non analizzabili le risposte a determinati segnali inviati dal computer. I dieschetti in programmi di questo tipo, sono solitamente perfettamente duplicabili, o al limite sono del tipo "livello 4".

Come proteggere i nostri programmi

Abbiamo visto i vari livelli di protezione conosciuti e raggiungibili. Proteggere non e' facile, specialmente quando a propria disposizione non si hanno strumenti adeguati.

A mio avviso indispensabile per una protezione che riguarda un programma Basic e' un efficace compilatore quale il PET SPEED, che ha il difetto di non accettare alcune istruzioni, quali i richiami a routines in linguaggio macchina, ed il pregio di rendre il programma compilato inattaccabile dai famigerati de-compilatori, che gia' operano sul lavoro dei programmatori.

Abbiamo parlato di routines di lettura diretta da disco: ecco un

listato che vi permettera' di scrivere e di leggere dati sul dischetto, evitando di informare la BAM di quanto presente sul supporto. Un consiglio: cercate di utilizzare le traccie esterne, per evitare che un eventuale inserimento di altri programmi come files pregiudichi l'informazione che immagazzinerete.

---------------- listato n.13 - legge scrive disco ----------------

Nel precedente paragrafo abbiamo anche trattato del controllo da parte del programma della presenza di errori sul dischetto.

Ecco la routine necessaria, avendo pero' l'accortezza di non utilizzarla troppo spesso durante lo svolgimento del programma, per evitare spiacevoli conseguenze alla testina ed alle orecchie dell'operatore...

------------------- listato n.14 - test errore --------------------

Alla linea 1100, che segue alla lettura del blocco, possiamo mettere un'istruzione che resetti il sistema se il codice dell'errore non e' quello desiderato; badate bene, pero', a mascherare la locazione di reset (64738), poiche' anche se il programma e' compilato, tramite un monitor e' analizzabile, ed una routine di questo tipo e' facilmente localizzabile. Provate ad assegnare delle variabili (ad es. A=12, B=15, ecc.) dove R = AxBxC+D/... in modo tale da mascherare efficacemente l'operazione tramite un semplice SYS R.

Se invece non vi importa nulla della copia del vostro programma, e tenete di piu' alla presenza del vostro nome sulla directory o in un punto preciso del dischetto (ad es. tramite un file USeR in traccia 18), e' opportuno che digitiate questa routine, che andra' a leggere, quando vorrete voi, l'intera directory, controllando il vostro nome o qualcos'altro, resettando o formattando il disco se qualcosa e' stato modificato.

------------------ listato n.14A- test directory ------------------

La maggior parte di voi si sara' fatta una bella risata: basterebbe "chiudere" il dischetto con una linguetta adesiva per impedire ulteriori scritture su di esso. Sta a voi impedire cio': il listato che segue va a scrivere dell'informazione sulla traccia 18 nel settore 18 (un settore solitamente mai usato dai titoli della directory); se il risultato e' un messaggio di errore, la routine avverte l'operatore che se il disco e' protetto il programma non gira. Il controllo puo' essere effettuato in qualsiasi momento, ed e' bene farlo sempre prima dell'apertura di un file (se il nostro e' un gestionale), o prima del caricamento del programma, se e' un gioco. Sta a voi decidere il resto.

-------------------- listato n.15 - test notch --------------------

Ed una volta scoperto che il nostro programma e' stato modificato, cosa facciamo ?
Se abbiamo accertato che il dischetto non e' protetto, le possibilita' sono infinite. Tramite le istruzioni di scrittura incontrate nella prima parte del libro, siamo in grado di operare sul

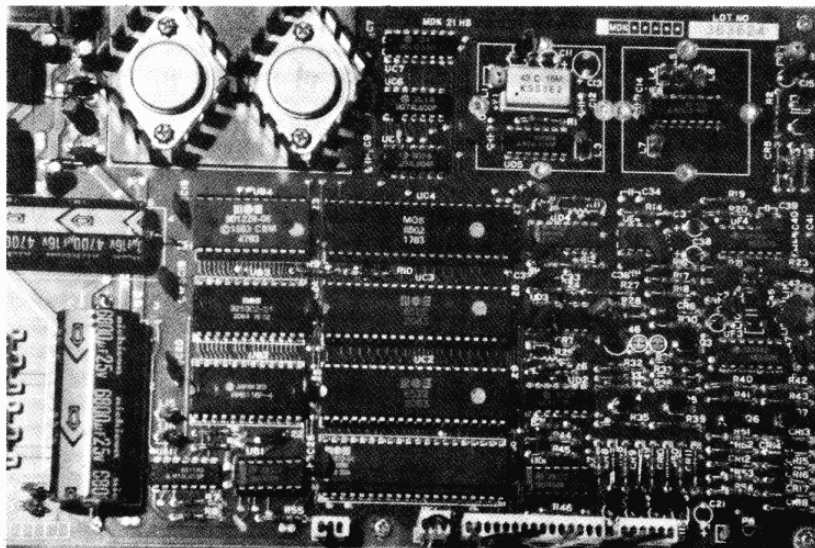
supporto in modo tale da cancellare totalmente l'informazione.

Possiamo anche utilizzare il programma creatore di errori per rendere il dischetto illeggibile, o solo per cancellare l'informazione conservando la directory (che diventerebbe completamente inutile).

Se pero' vogliamo fare gli originali, ecco un programmino da caricare e da far funzionare, che ugualmente cancella l'informazione, aggiungendo un tocco personale...

Restando nei limiti della creanza, oppure no, sono molti i messaggi amichevoli da indirizzare ad un pirata del software...

-------------------- listato n.16 - destroy 1.0 --------------------

ESPLOSO INTERNO DRIVE 1541

Capitolo 6: I copiatori

Abbiamo parlato delle possibili protezioni che un programma puo' avere, ma dobbiamo tener conto del fatto che anche noi possiamo avere la necessita' di copiare del software, sia per non rischiare di rimanere con una copia unica (che anche un granello di polvere puo' rovinare), sia per utilizzare il software che qualche nostro amico vuole donarci.

La legge italiana, oscura per quanto riguarda un problema che ancora non e' stato affrontato se non marginalmente, punisce coloro che duplicando il software, non ne denuncia la vendita, frodando il fisco e danneggiando lo Stato.

Non e' reato duplicare il software, ma venderlo senza autorizzazione, si'.

I modi possibili per copiare sono principalmente due: copiare i singoli files, trasferendoli ad uno ad uno oppure ricopiare fedelmente l'intero disco, in tre passaggi. Il computer non ha problemi, anche se si tratta di programmi contenenti un autostart o in linguaggio macchina. I copiatori di files, cosi' come quello qui di seguito presentato, in linguaggio macchina, utilizzano la memoria nel modo seguente: il copiatore non fa altro che riconfigurare la memoria interna del computer, caricare in una locazione diversa dall'originale il programma (memorizzando le locazioni originali) e li salva nelle stesse posizioni sul disco copia.

Ecco come la memoria viene deformata:

MEMORIA DEL C-64 ALL'ACCENSIONE

ESAD.		DEC.
FFFF		65535
	8K ROM KERNAL	
E000		57344
	(CARATTERI ROM) 4K I/O	
D000		53248
	4K RAM (BUFFER)	
C000		49152
	8K BASIC ROM	
A000		40960
	8K RAM (O CARTUCCIA)	
8000		32768
	16K RAM PROGRAMMATORE	
4000		16384
	14K RAM	

	INDIRIZZI BASSI DEL SISTEMA	
0000		0000

Dove la memoria compresa tra D000 e C000 comprende anche:

I/O memoria del disco
I/O espansione (CP/M)
CIA # 2 memorie
CIA # 1 memoria
Colore reg. RAM
Memoria SID
Memoria VIC

Negli indirizzi bassi del sistema, troviamo anche la MEMORIA DI SCHERMO, compresa tra 0400 e 0800 (1024-2048 dec.) e 1K dedicato alla zona di controllo con la pag. ZERO.

La disponibilita' dell'operatore: 38 Kbytes.

1a CONFIGURAZIONE

Questa particolare disposizione della memoria viene usata prevalentemente dai copiatori di files, che utilizzano routines facenti riferimento al KERNAL.

Indirizzo	Contenuto
FFFF	
	8K ROM KERNAL
E000	
	4K I/O
D000	
	4K RAM
C000	
	16K RAM
8000	
	16K RAM
4000	
	16K RAM
0000	

DISPONIBILI 52 K

RAM

2a CONFIGURAZIONE

Questa viene utilizzata da copiatori evoluti, che non avendo bisogno del KERNAL riescono ad ottenere quasi l'intera memoria disponibile.

```
FFFF ____________________________
           8K RAM
E000 ____________________________
           4K I/O
D000 ____________________________
           4K RAM
C000 ____________________________        DISPONIBILI 60 K
          16K  RAM
8000 ____________________________                RAM
          16K  RAM
4000 ____________________________
          16K  RAM
0000 ____________________________
```

Ecco il listato in Basic di due copiatori che trasferiscono un programma dal nastro al disco e dal disco al nastro. Essi tengono conto dell'inizio del file, memorizzano l'informazione, e la riscrivono su un diverso supporto.

---------------------- listato n.17 - copia n/d ----------------------

---------------------- listato n.18 - copia d/n ----------------------

Occorre ricordare che se un programma e' su piu' files, occorre, modificando il supporto, modificare anche il numero di periferica usata se c'e' una concatenazione di informazioni.

--------------------- listato n.19 - copia-files ---------------------

3a CONFIGURAZIONE

Configurazione preferita dai copiatori totali di dischetti: al C=64 non viene richiesta alcuna operazione, ma solo di rendere disponibile l'intera sua memoria come recipiente di passaggio, per permettere il trasferimento dell'intera informazione presente su un disco, su un altro supporto.

Ricordate quanti blocchi sono presenti su un disco? Sono 683, dove ognuno e' formato da 256 bytes (1K ogni 4 blocchi), ed il drive riesce a trasportare, velocemente, tutti i dati in tre passaggi. Il lavoro viene svolto esclusivamente dal DOS del 1541, mentre la tastiera del computer viene disabilitata (viene riconosciuta solo la pressione di un tasto qualsiasi).

```
FFFF ____________________________
          16K RAM

C000 ____________________________
          16K RAM             MEMORIA DISPONIBILE

8000 ____________________________       64 K RAM
          16K RAM
4000 ____________________________

          16K RAM
0000 ____________________________
```

Non esiste piu' KERNAL (per chi non lo sapesse, e' un insieme di routines utilizzabili da Basic tramite i comandi SYS), interprete Basic, la possibilita' di controllare periferiche da tastiera, ecc.

Ecco un copiatore in linguaggio macchina che si trasferisce interamente nel DOS del 1541, programmandolo in modo tale da sfruttare il C=644 come un catino...

-------------------- listato n.20 - copia-disco --------------------

E' un copiatore "evoluto"; in caso di errore, non leggera' l'informazione e continuera' dal blocco successivo, visualizzando traccia e settore interessati.

Ancora sulla ri-configurazione

Una delle migliori del C=64 e' il modo in cui la sua memoria puo' essere ri-configurata. Abiamo visto come questo sia possibile dato che il C=64 possiede 64 K di memoria compresi tra le locazioni 0000 e FFFF. Nella prima tabella contenente la memoria del computer all'accensione, notiamo che vi sono delle aree di RAM sotto quelle utilizzate dalla ROM e dagli I/O.

Il 6510 permette al programmatore di spegnere dei banchi di memoria; nel caso dele aree Basic e KERNAL, possono avere al di sotto delle aree di RAM, ma esse possono essere soltanto lette, se la ROM e' inserita. Per scrivere il contenuto di queste aree, occorre disinserire la ROM, e di conseguenza dovremo fare a meno dell'interprete Basic. Ecco perche' la maggior parte dei copiatori utilizzano l'Assembly. Usando anche l'area di I/O (Input/Output = Ingresso/Uscita), il programma non deve contenere vettori di interruzione, non deve avere istruzioni in Basic, e non deve utilizzare routines KERNAL. Oltre a queste limitazioni, le possibilita' di riconfigurazione della memoria del computer fanno del C=64 un elaboratore che e' superiore agli altri a 8 o 16 bits.

Capitolo 7: L/M ed il 1541

Un ultima considerazione sui copiatori

Molti copiatori utilizzano i sistemi sopra descritti, ma non vorrei che ci fermassimo qui. La programmazione puo' essere aiutata da queste tecniche, e mi sembra sciocco parlarne solo in termini di duplicazione. D'altronde, non ci stiamo occupando di linguaggio macchina per la programmazione del C=4, ma dell'utilizzo di routine programmanti il DOS del 1541.

Non manchera' probabilmente l'occasione di approfondire, magari in un'altra pubblicazione, quello che sembra un argomento a molti oscuro, e che invece e' semplicemente un linguaggio (come il Basic) da imparare nel modo piu' facile che esista.

------------------ listato n.21 - copia errore -------------------

Il 1541

Il nostro drive, come abbiamo visto, e' un'unita' a dischi intelligente, dotata di un Sistema Operativo grazie ad un miniprocessore.

In poche parole, una volta che abbiamo dato le istruzioni per operare, la memoria centrale viee ignorata, ed e' il DOS a lavorare. Sono tre le operazioni eseguite dal drive contemporaneamente:

1 - controlla il traffico dei dati tra il C=64 e se stesso.

2 - interpreta le istruzioni e le esegue manipolando i files, aprendo e chiudendo canali di comunicazione, gestendi i buffers disponibili.

3 - esegue una serie di operazioni miranti a rendere riconoscibile un supporto magnetico estraneo (come la formattazione, scrittura e lettura di files, ecc.), chiamate dagli statunitensi "hardware-oriented".

Tutto cio' e' possibile grazie al lavoro del 6502 ed alle tecniche di Interrupt. Un Interrupt e' la ricezione di un comando da parte del 1541, ricevuta tramite un impulso su un canale aperto.
Il DOS, automaticamente, interrompera' (da qui il nome) il suo lavoro, registrera' il comando, finisce il lavoro e poi si predispone ad accogliere istruzioni.

L'esecuzione e' quella logica, e si limita ad aprire e chiudere canali. Le altre operazioni, invece, hanno bisogno dell'esclusivo lavoro del processore del drive.
Un temporizzatore interrompe ogni 14 millisecondi l'esecuzione dei normali programmi, e la comunicazione tra due programmi diversi e' garantita da una comune area di memoria.

Le porte di I/O

VIA = Versatile Interface Adapters

```
--------------------------------------
VIA 6522 1, Porta seriale BUS

   $1800      Porta B
   $1801      Porta A
   $1802      DIREZIONE DATI B
   $1803      DIREZIONE DATI A
   PB 0    :    DATI IN ENTRATA
   PB 1    :    DATI IN USCITA
   PB 2    :    CLOCK INSERITO
   PB 3    :    CLOCK DISINSERITO
   PB 4    :      ATN A
   PB 5,6  :    INDIRIZZO DEVICE
   CB 2    :      ATN IN
--------------------------------------

--------------------------------------
VIA 6522 2, Porta motore e Controllo
            Lettura e Scrittura Testina

   $1C00      Cotrollo Porta B
   $1C01      Dati in lettura/scrittura
              testina per porta A
   $1C02      DIREZIONE DATI A
   $1C03      DIREZIONE DATI B
   PB 0    :    STP = I (Stepper motor)
   PB 1    :    STP = O (mov. test.)
   PB 2    :    MTR (pilotaggio motore)
   PB 3    :    ACT (accensione LED)
   PB 4    :    WPS SWITCH
                (protezione scrittura)
   PB 7    :    SYNC (sincronizzazione)
   CA 1    :    Byte pronto
   CA 2    :    SOE
--------------------------------------
```

La mappa di memoria

Ecco una mappa della memoria (semplificata) del 1541, e qui di seguito vedremo in dettaglio le cose piu' interessanti riguardo ad essa.

La ROM residente e' di 16 K, ed e' provvista di un unico sistema operativo (DOS= Disk Operating System) dove la maggior parte delle funzioni e' realizzata tramite software; grazie cio' si e' potuto progettare un tipo di hardware relativamente costoso con un software a prezzi decisamente accessibili. Lo scopo della Commodore, infatti, e'stato quello di produrre un drive poco costoso abbisognante di un software il piu' eclettico possibile.

Volendo esaminare l'intero DOS, occorre tramite un drive-monitor che carichi l'intera memoria e la trasferisca nel computer (una

successiva istruzione permettera' la stampa dell'intero disassemblato). Avremmo potuto pubblicare anche noi pagine e pagine di linguaggio macchina (perche' 16 K di memoria sono molti), ma per la maggior parte di voi sarebbe stato inutile.

Come utenti, e' possibile utilizzare la ROM tramite dei salti (JSR) nel suo interno, mentre coloro che sanno qualcosa di linguaggio macchina e di ri-configurazione di memoria hanno la possibilita' di bruciare (cioe' di escludere) un altro set di ROM con qualche paricolae routine modificata, e tutto cio' per poter scrivere in modo anormale impedendo la lettura da parte dei normalo copiatori.

Riuscendo a far cio' senza interferire nei normali schemi, potremo ottenere un'ottima protezione, dato che l'unica contro-misura consisterebbe nella scrittura di una routine che riproduca il nostro schema, inserita nella RAM non usata del drive, che possa essere in grado di scrivere e di leggere allo stesso modo.

Ma il maggior problema che si incontra nella stesura di questo tipo di routines e' l'esiguo spazio di RAM disponibile. L'area di ROM compresa da $C00 e $F24C viene utilizzata maggiormente per manipolare dati, interpretarli, e cosi' via.

Meglio quindi non modificarla. La ROM compresa tra $F24D e $FFF contiene le routines di controllo dell'hardware; esse controllano cose come la selezione di blocchi e di traccie, la lettura e la scrittura, e cosi' via. Gli inserimenti di errori possono essere fatti tramite questa zona, o modificandola.

La RAM utilizzabile dall'operatore ha un'estensione di 2 K: abbiamo una pagina zero (richiesta dal 6502) da $0000 a $00FF, due pagine da $0100 a $02FF sono riservate ai puntatori addizionali, accodamenti ed altro. Nell'area compresa tra $0250 e $02B0 vi sono i vettori della RAM, e si gestisce il trattamento dei files. Da $02b1 a $02FF viene gestita l'informazione in uscita, come messaggi di errore e visualizzazione della directory; tutta l'area di RAM rimanente e' divisa in buffers di 256 bytes ciascuna.

Ci sono 5 buffers che caricano i dati dal disco e quelli ad esso destinati; vengono posizionati secondo le esigenze del DOS. Proprio in questa zona vengono memorizzati i programmi dell'utente, dove i primi 18 bytes del buffer ($0500) vengono solitamente usati per saltare alla routine inserita dal programmatore.

L'area piu' importante nella RAM e' quella compresa tra le locazioni $0000 e $0005, e tra $0006 e $0012, dove si provvede al codice dei lavori eseguiti ed alla memorizzazione degli headers (traccia e blocco) del lavoro in codice.

Dopo la ROM e la RAM, parliamo di due Versatile Interface Adapters (VIA), interpretati dalla CPU come locazioni di memoria. Lo schema ve l'ho gia' presentato precedentemente.

Un VIA occupa 16 locazioni di memoria: il primo VIA e' compreso tra $1800 e $180F; controlla il bus seriale agisce da tramite dal bus interno verso l'esterno. Il secondo, invece, racchiuso tra $1C00 e $1C0F ha come compito il controllo dell'hardware: e' infatti collegato con il circuito di azionamento del motore, la verifica della cellula della linguetta adesiva (notch) ed il controllo della logica di scrittura e lettura per le testine. I segnali hardware sono trasmessi direttamente dal buds interno dei dati del drive. Inoltre, come descritto nelle tabelle sopra descritte, i VIA dispongono di temporizzatori indirizzabili (CLOCK) a seconda delle esigenza del DOS.

Altra caratteristica importante e' la possibilita' di "specchiare" la memoria (MIRRORING): le parti componenti la memoria possono essere viste in diverse posizioni all'interno della mappa.

Ecco un listato, in due parti, una Basic che funge da caricatore, ed un'altra in linguaggio macchina che si trasferisce nel DOS, che ci permettera' di modificarlo in modo tale da formattare un disco in meta' tempo!

------------------ listato n. 22 - formatta veloce ------------------

La Memoria-Disco (i principali indirizzi)

0	$00	Codice comando BUFFER 0
1	$01	Codice comando BUFFER 1
2	$02	Codice comando BUFFER 2
3	$03	Codice comando BUFFER 3
4	$04	Codice comando BUFFER 4
6	$06-07	Traccia e Blocco BUFFER 0
8	$08-09	Traccia e Blocco BUFFER 1
10	$0A-0B	Traccia e Blocco BUFFER 2
12	$0D-0D	Traccia e Blocco BUFFER 3
14	$0E-0F	Traccia e Blocco BUFFER 4
18	$12-13	Numero ID per drive 0
20	$15-15	Numero ID per drive 1
22	$16-17	ID
32	$20-21	FLAG movimento testina
48	$30-31	Puntatore del BUFFER per il controllore Disco
57	$39	Costante 8, inizio dati dell' Header del blocco
58	$3A	Parita' dati BUFFER
61	$3D	Numero Drive per controllore
63	$3F	Numero BUFFER per controllore
67	$43	Numero blocchi per traccia durante la formattazione
71	$47	Costante 7 , inizio dati dell' Header del blocco
73	$49	Puntatore Stack
74	$4A	Conta passi della testina
81	$51	N. traccia durante la formatt.
105	$69	Ampiezza del blocco
106	$6A	Numero tentativi lettura
111	$6F-70	Puntatore indirizzo per i comandi B e M.
119	$77	N. di periferica (LISTEN)
120	$78	N. di periferica (TALK)
121	$79	Flag (LISTEN)
122	$7A	Flag (TALK)
124	$7C	Flag per ATN (rceve dati dal BUS seriale)
125	$7D	Flag per EOI dal BUS seriale
127	$80	Numero della periferica
128	$80	Numero della traccia
129	$81	Numero del blocco
130	$82	Numero del canale aperto

131	$83	Indirizzo secondario
132	$84	Indirizzo secondario
133	$85	Byte di dati
139	$8B-8D	Area di immagazzinamento div.
148	$94-95	Puntatore attuale BUFFER
153	$99-9A	Indirizzo del BUFFER 0 ($300)
155	$9B-9C	Indirizzo del BUFFER 1 ($400)
157	$9D-9E	Indirizzo del BUFFER 2 ($500)
159	$9F-A0	Indirizzo del BUFFER 3 ($600)
161	$A1-A2	Indirizzo del BUFFER 4 ($700)
163	$A3-A4	Puntatore BUFFER Input $200
165	$A5-A6	Puntatore BUFFER per imessaggi di errore
181	$B5-BA	Record basso, Blocco basso
187	$BB-C0	Record alto, Blocco alto
193	$C1-C6	Puntatore creazione file relat.
199	$C7-CC	Lunghezza del record (f.rel.)
212	$D4	Puntatore record file relativo
213	$D5	Numero di blocchi interessati
214	$D6	Puntatore blocco dati
215	$D7	Puntatore record del file rel.
231	$E7	Tipo di file
249	$F9	Numero del BUFFER
256-325	$100-145	Area STACK
512-552	$200-228	BUFFER stringa del comando
586	$24A	Tipo di file
600	$258	Lunghezza del record
601	$259	Traccia del settore interessato
602	$25A	Settore interessato
628	$274	Lunghezza linea di Input
632	$278	Numero dei nomi di files
633	$279	Metodo di controllo dei files
640-644	$280-284	Traccia di un file
645-649	$285-289	Blocco di un file
725-761	$2D5-2F9	BUFFER a disposizione per i messaggi di errore
762-764	$2FA-2FC	Numero di blocchi liberi

768-1023	$300-3FF	BUFFER 0
1024-1279	$400-4FF	BUFFER 1
1280-1535	$500-5FF	BUFFER 2
1536-1791	$600-6FF	BUFFER 3
1792-2047	$700-7FF	BUFFER 4

E per finire con l'utilizzo del linguaggio macchina nella programmazione del 1541, ecco un listato, sempre in due parti, che dimezzera' il tempo di caricamento dei programmi, applicando un "turbo" al DOS del nostro drive. In pratica non viene aumentata la velocita' del motorino, ma semplicemente modificata la quantita' di dati leggibile nell'unita' di tempo.

------------------ listato n. 23 - carica veloce------------------

Capitolo 8: Editor per 1541

Che cos'e' un Editor

Una delle migliori forme di scrittura diretta su disco e' realizzabile tramite l'uso di un Editor; non e' altro, infatti, che un'utility che ci permette di accedere in modo ragionato ed analitico al blocco di una traccia qualsiasi del disco, dandoci l'opportunita' di modificarlo, cancellarlo o riscriverlo completamente.

Un blocco e' formato da 256 bytes, formanti una tavola ideale di 16x16 cellette di memoria.
Avrete gia' notato quanto sia ricorrente il numero 16 quando si parla di locazioni di memoria in esadecimale. Per chi non lo sapesse, credo poche persone tra voi, vi sono due scale di valori che interessano il nostro computer: i valori decimali, corrispondenti al nostro comune modo di contare, ed i valori esadecimali, che invece di andare di 10 in 10, variano di 16 in 16. Ecco un esempio pratico:

DECIMALE

00,01,02,03,04,05,06,07,08,09,10,11,12,13,14,15..

ESADECIMALE

00,01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F..

Per la conversione da decimale ad esadecimale, solitamente, si fa uso di tavole apposite, troppo lunghe per essere pubblicate, oppure si applica un particolare algoritmo, che e' stato inserito nel seguente listato, l'Editor appunto, che vi facilitera' di molto il lavoro.

--------------------- listato n.24 - Editor ----------------------

Usando questo tipo di programma occorre fare molta attenzione: se non si e' capaci di utilizzare al meglio le sue capacita' e' possibile perdere in modo irreversibile.

Il listato presenta tutte le istruzioni di lettura e di scrittura che abbiamo incontrato nella parte iniziale di questo libro, ed e' un'ottima pratica per la programmazione avanzata su disco.

Le istruzioni

Al momento del RUN, il dischetto verra riportato sulla traccia 18 tramite un Inizialise. Dopo qualche istante, premendo un tasto qualsiasi, sarete in grado di visualizzare la schermata principale dove si svolgera' la gran parte del lavoro.

Ecco i comandi, seguiti dalla loro funzione:

- @ -

↑ ↑ ↑ Questo tasto permette all'operatore di modificare il byte posto al di sotto del cursore. Non e' una modifica diretta, ma viene richiesto il codice DECIMALE, che verra' poi convertito direttamente dal programma. Ricordate che quest'operazione avviene, per il momento, SOLO nella memoria del computer. Il blocco non sara' modificato fino a quando non daremo l'istruzione per scriverlo sul disco.

- N -

↑ ↑ ↑ L'istruzione, utilissima per capire dove viene memorizzato su disco il programma, permette la lettura del blocco successivo riferentesi al file il cui settore e' al momento in memoria. Ad esempio, se un programma e' memorizzato a partire dal 1o blocco della 17esima traccia, e prosegue nel 4o settore (della stessa traccia), avremo l'opportunita' di consultarlo direttamente saltando i blocchi intermedi.

- J -

↑ ↑ ↑ Salta al blocco indicato dalla posizione del cursore. Se nessun blocco viene indicato, viene riletto il blocco precedente, e cancellato il lavoro fino a quel momento svolto.
Atenzione, quindi, ad usarlo.

- Q -

↑ ↑ ↑ Permette di uscire dal programma senza modificare il blocco interessato. E' da usare dopo aver memorizzato il blocco, o prima se ci siamo accorti di aver sbagliato a modificare il settore.
Sono presenti delle routines di controllo che permettono all'operatore di modificare all'ultimo momento il comando dato, in caso di errore.

- R -

↑ ↑ ↑ Il comando piu' importante, ed il piu' pericoloso da usare senza accorgimenti: riscrive il blocco corretto sul dischetto contenuto dal drive.
Prima di utilizzarlo, e' meglio che impariate quale sia la struttura del blocco di dati che state analizzando.

- B -

↑ ↑ ↑ Permette la lettura di un altro blocco, di cui dovrete

inserire numero di traccia e di settore. Una routine impedira' l'errore, abbastanza facile, di lettura su blocchi o traccie inesistenti.
Inutle dire che al momento della lettura, il supporto DEVE essere gia' stato formattato, altrimenti non potra' essere letto (lo stesso vale se e' presente un errore...)

- I -

↑ ↑ ↑ Grazie a questa istruzione e' possibile inserire una stringa di testo all'interno del blocco. E' un comando molto usato per la modifica di testi (su file sequenziali) o per cambiare i nomi dei programmi inseriti nella directory. Tenete conto, pero', che molti programmi hanno un caricamento concatenato, il che vi costringera' a modificare anche l'istruzione all'interno del caricatore.

- +/- -

↑ ↑ ↑ ↑ Premendo questi due tasti e' possibile leggere il blocco successivo (+) o quello precedente (-) al settore che appare in questo momento sul video.
Cambiando traccia, pero', sara' necessario premere la funzione "B" per saltare nelle giuste locazioni di memoria.

Valore decimale

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ Indica in qualsiasi momento il valore decimale del byte presente sotto il cursore ed e' l'ideale per chi non ha molta confidenza con il Codice ASCII.

Posizione

↑ ↑ ↑ ↑ ↑ Calcola l'esatta posizione del byte sotto il cursore; vedremo quanto utile e preziosa sara' questa indicazione, che ci permettera' di operare con una certa sicurezza modificando i blocchi.La posizione e', naturalmente, calcolata i valori decimali, ed e' a questi che faremo riferimento nelle nostre spiegazioni.

Un'ultima raccomandazione

Per i primi tempi lavorate sempre su una copia del disco che avete. E' possibile, ripeto, sbagliare e non poter piu' correggere l'errore. Se avete intenzione di modificare solo la directory per i nomi in essa contenuti, o l'intesatzione del disco, vi consiglio di utilizzare le utility gia' presentate o quelle comunque presenti in questo libro, che garantiscono la perfetta esecuzione dell'istruzione.

Vedremo, nel prossimo capitolo, quante cose sia possibile fare grazie ad un editor.

Capitolo 9: Proteggere in Editor

Proteggere un programma non e' facile. Abbiamo visto a che punto si e' arrivati in questo tipo di tecniche. Ma saper proteggere equivale anche al saper "sproteggere"...

Quali sono le reali capacita' in nostro possesso, come modificare il disco, personalizzarlo, impedire la visione della directory, inserire un messaggio, caratterizzare la directory, ecc.? Con un po' di pazienza affronteremo tutte queste problematiche, alla ricerca della praticita' e del buon gusto (se e' possibile...)

La traccia 18, blocco Ø

Questa traccia e' fondamentale, ed e' da considerarsi di gran lunga la piu' importante.

Carichiamo il blocco Ø, ed analizziamolo: abbiamo dinanzi la BAM, e grazie alla conversione decimale che abbiamo, siamo in grado di ragionare in termini molto piu' semplici.

Partendo da sinistra, troviamo il numero 18 (badate bene di osservare non tanto il carattere che appare nel blocco, quanto la conversione decimale in fondo alla maschera grafica di schermo) che rappresenta la traccia successiva dove e' memorizzato il resto dell'iformazione. Il byte successivo, equivalente ad 1, rappresenta il blocco successivo di lettura. In poche parole, la lettura da parte del DOS inizia dalla traccia 18 blocco Ø, e continua in traccia 18 blocco 1, e cosi' via. Questo vale PER TUTTI I BLOCCHI, che utilizzano queste due locazioni esclusivamente per questo.

La terza locazione di memoria e' molto importante: e' visualizzata una "A", 65 decimale, e rappresenta un punto di riferimento per il DOS. Se noi andiamo a modificare questa lettera, avremo il nostro dischetto permanentemente protetto in scrittura. Potremo, cioe', leggere ma non scrivere dati. Il messaggio di errore "73, DOS MISMATCH,18,ØØ" indica che il dischetto non e' compatibile con il drive. Questo perche' la Commodore produce diversi tipi di drive, che tra loro sono completamente compatibili in lettura ed in scrittura, o solo in lettura. Per rendere di nuovo compatibile il dischetto, e' necessario ri-formattare completamente il supporto, cancellando tutta l'informazione compreso il 3o byte.

Il quarto byte, invece, non ha alcuna importanza, e solitamente ha un valore Ø.

Le successive 14Ø locazioni contengono la BAM, ed ogni gruppo di 4 bytes descrive i blocchi disponibili per ogni traccia. Il primo valore (ad es. 21) ci dice quanti blocchi siano disponibili nella prima traccia, ed i 3 bytes successivi sono l'analisi numerica di quanto detto prima. In esadecimale ci apparirebbero i seguenti valori:

```
15 FF FF 1F
```

dove 15 e' uguale a 21 in esadecimale, e il resto dei valori ha il seguente significato:

```
15    F    F    F    F    1    F
```

15 FF FF 1F

dove 15 e' uguale a 21 in esadecimale, e il resto dei valori ha il seguente significato:

```
15     F      F      F      F      1      F

↑↑     ↑↑     ↑↑     ↑↑     ↑↑     ↑↑     ↑↑
__    ____   ____   ____   ____   ____   ____
21    1111   1111   1111   1111   0001   1111
```

Come vedete, la somma dei blocchi liberi e' appunto 21.
Un altro esempio: ecco la traccia 35

```
11     F      F      F      F      0      1
```

dove 11 equivale al decimale 17, F e' uguale a 4 blocchi liberi, 0 a nessun blocco e 1 ad un blocco disponibile.

La BAM viene aggiornata ogni volta che operiamo dei cambiamenti alla directory, tramite comandi come SAVE, SCRATCH, VALIDATE, ecc.

I bytes compresi tra il 144 e il 161 contengono il nome del disco, che deve essere al massimo lungo 16 caratteri. Se il nome risulta piu' lungo, esso viene automaticamente ridotto dalla scrittura di spazi vuoti.

I bytes 162 e 163 racchiudono la I.D. del dischetto, che viene riportato su ogni header di blocco del supporto.

Il byte 164 e' uno spazio shiftato (A0 in esadecimale).

I bytes 165 e 166 sono la rappresentazione in Codice ASCII della scrittura 2A, simbolo del tipo di DOS e di formattazione.

I bytes 167, 168, 169 e 170 sono spazi shiftati (questo carattere viene utilizzato per riempire tutti gli spazi vuoti che si vengono a creare sul disco).

I bytes seguenti, dalla 171 alla 255 locazionesono azzerati, ma ogni tanto e' possibile trovarvi dei caratteri ASCII.

La BAM e' modificabile in un numero infinito di modi: basta avere un po' di fantasia e non utilizzare un dischetto con programmi la cui perdita potrebbe sconvolgerci... almeno per imparare.
In ogni caso, vi consiglio di effettuare un cambiamento alla volta , per poi listare la directory e vedere gli effetti del nostro operato. L'uso dei caratteri ASCII e' fondamentale, e vi consiglio di impararne bene la tabella.

Vediamo byte per byte cosa e' possibile modificare.

I primi 2 bytes (0 e 1) contenenti i puntatori del blocco successivo, e 'meglio non modificarli, altrimenti non risultera' piu' leggibile la directory.

Il byte n.2 puo' essere cambiato, ma una volta fatto dovrete rassegnarvi a non poter piu' scrivere sul dischetto (sara' come se aveste messo un'etichetta adesiva perenne sul taglio laterale. Alcuni copiatori, poi, potrebbero essere messi in difficolta' da questa locazione. Infatti, se sono necessari tre passaggi per copiare un intero disco, al secondo verra' bloccato il processo di scrittura dall'intestazione di questo blocco.

Il byte 4 raccoglie i blocchi liberi della prima traccia, e se

noi modifichiamo questa locazione il DOS non utilizzera' piu' i blocchi che secondo lui risultano occupati. Ogni 4 byte c'e' l'inizio di un'altra traccia memorizzata dalla BAM.

I bytes dal 5 al 7 sono la mappa in bit dei blocchi ancora allocabili:

1 = disponibile
Ø = allocato (e percio' non disponibile)

Cambiando i valori di queste locazioni, confondiamo il DOS su quale blocco sta lavorando, ma tutto puo' essere riportato alla normalita' da una VALIDATE.

I bytes dall'8 al 143 rappresentano in gruppi di 4 tutte le traccie presenti sul dischetto.

I bytes dal 144 al 161 racchiudono il nome del disco, e questo puo' essere cambiato a vostro piacere, inserendo anchedei caratteri che risulterebbero illegali tramite un comando diretto (ad es. l'asterisco o il punto interrogativo). Uno dei modi piu' classici consiste nell'uso del carattere DELETE (14 esa., 2Ø dec.), che posto in sequenza all'intestazione del disco occupandone le ultime 6 locazioni cosi':

2Ø 2Ø 2Ø ØØ ØØ ØØ

impedisce la visualizzazione della directory. Attenzione, stiamoparlando di valori DECIMALI; infatti il valore 2Ø in esad. corrisponde allo spazio.

I bytes successivi contenenti la I.D. possono essere modificati a piacere senza rovinare nulla (162 e 163), anche se non verranno cambiati gli I.D. di tutti i blocchi del dischetto.

Le locazioni 165 e 166, dove risiede la designazione della formattazione, possono essere modificate, a patto che i caratteri siano leggibilidal DOS, per non creare problemi di lettura successiva della directory.

Nella zone seguente (167-255) qualsiasi variazione non sortisce alcun effetto sul contenuto del disco ed e' un'ottima area per inviare messaggi ai pirati di software.

La traccia 18, blocco 1

A questo punto passiamo ad analizzare il blocco successivo, dove cioe' inizia la directory del nostro dischetto.

La directory puo'contenere un massimo di 144 files, ed in essa possiamo trovare tuttele informazioni che ci possono interessare riguardante i files. Ma vediamo analiticamente quale possa essere il contenuto del blocco 1 di traccia 18.

I bytes Ø e 1 contengono sempre ipuntatori al blocco successivo (traccia e settore); nel caso la directory contenga 8 files o meno, i due puntatori risulteranno essere Ø e 255 (ØØ e FF in esadecimale), locazioni che avvertono il DOS che questo e' l'ultimo settore da leggere di un file.

Il blocco sara' cosi' strutturato:

byte 2 - byte 31 .... file n. 1
byte 34 - byte 63 .... file n. 2
byte 66 - byte 95 .... file n. 3
byte 98 - byte 127 .... file n. 4
byte 130 - byte 159 .... file n. 5
byte 162 - byte 191 .... file n. 6
byte 194 - byte 223 .... file n. 7
byte 226 - byte 255 .... file n. 8

Nelle locazioni indicate troveremo tutte le indicazioni riguardanti un file che ci interessa, e cioe' il tipo, dove si trova, il suo nome e la sua estensione in blocchi. Ma analizziamo ancora una volta analiticamente il contenuto di questo blocco.

I bytes 0 e 1 indicano il successivo blocco della directory, che e' in traccia 18, settore 4 (e questo perche' viene memorizzata saltando tre blocchi alla volta). Nel caso non vi siano altri blocchi da leggere, i valori sarebbero 0 e 255; volendo modificare le locazioni, se in esse memorizziamo i valori 18 e 1, otterremo un continuo segnale di caricamento dopo il comando di lettura della directory: il puntatore continuerebbe a tornare su se stesso, creando un loop infinito. L'unico modo per fermarlo consiste nello spegnere il drive o il computer.

Riportando pero' i valori originali e' possibile ripristinare la funzionalita' di queste locazioni.

Il byte 2 contiene le informazioni che i DOS utilizza per classificare il tipo di file; ecco i valori:

Esad.	Decim.	Significato
00	00	Nessun file o file cancellato
80	128	Nessun file o file cancellato
81	129	File SEQuenziale
82	130	File PRoGramma
83	131	File USeR
84	132	File RELativo

Questo byte non deve essere modificato, a meno che non vogliate confondere le idee a qualche pirata di software. Cambiarlo significa il piu' delle volte cancellare il file dalla directory.

I bytes 3 e 4, invece, contengono i puntatori del primo blocco e traccia da cui ha inizio il file. Da non modificare.

I bytes dal 5 al 20 hanno memorizzato il nome del file, che deve essere lungo 16 caratteri (ma in caso fosse piu' corto, il DOS provvede a riempire gli spazi vuoti con degli spazi shiftati).
Usando dei caratteri non visibili da Basic e' possibile camuffare il nome del file, ma e' una tecnica che scoraggia solo i principianti. La maggior parte dei copiatori di files, funzionanti in linguaggio macchina, sono lo stesso in grado di richiamarli, e di conseguenza di copiarli.
E' possibile, inoltre, inserire caratteri particolari quali il clear/home, o il delete, facendo riferimento sempre alla tabella ASCII. Nel caricarli da Basic, tenete conto di quanto avete fatto, ed

al limite utilizzate la forma di caricamento con il nome del file abbreviato (scrivendo la prima parte del nome seguita da un'asterisco).
Per personalizzare ulteriormente la directory, possiamo scrivere addirittura il numero di device dopo il titolo del programma, in modo che venga visualizzato ad esempio:

```
10 "START",8,1    PRG
```

Basta lasciare uno spazio shiftato dopo la fine del nome, dopo il quale possiamo scrivere quel che vogliamo senza modificare il file.
I bytes da 21 a 23 sono solitamente non utilizzati, e vengono riempiti solo da files relativi.
I bytes 28 e 29 sono nulli.
Il resto della directory e' semplicemente la ripetizione per ogni file. Ricordatevi che la modifica della directory ha piu' una funzione estetica che "protettiva", poiche' solo i principianti vengono fermati da questo tipo di difficolta'.

Capitolo 10: Test sulla meccanica

Ricapitolando

Vediamo di riassumere in poche righe le caratteristiche tecniche del nostro 1541:

Dimensioni

Altezza	mm	97
Larghezza	mm	200
Profondita'....	mm	374

Caratteristiche elettriche

Frequenza	Hz	50
Tensione	Volt	220
Assorbimento ..	Watt	25

Tipo di memoria

Capacita' dei dischi	174.848 bytes
Num. di files max.	144
Num. settori per traccia ...	17 - 21
Num. bytes per blocco	256
Numero traccie	35
Numero blocchi totale	683
Numero blocchi disp.	664

La testina magnetica

Uno dei maggior problemi che si possono incontrare usando il 1541 e' la facilita' estrema al disallineamento della testina. Questo non fa altro che creare problemi di lettura su dischi formattati da altri drives. Ma cosa s'intende per disallineamento?

Specialmente i primi modelli del 1541, quelli ancora color latte, ed i primi di quelli piu' scuri, sono stati costruiti per leggere dati in una certa maniera, diciamo cosi' tranquilla. Ma dopo un anno dall'uscita di questi lettori, e' sorta la moda di proteggere con errori il supporto magnetico, creando una sollecitazione maggiore di tutta la meccanica.

Abbiamo gia' parlato di quel che succede nel momento in cui un errore viene riscontrato sulla superficie del supporto: la testina viene ri-inizializzata, picchiando contro il fermo di acciaio posto al limite esterno del cerchio dove poggia il dischetto.

La funzione di questo piccolo trapezio consiste nel fissare l'inizio della zona formattabile del dischetto. La testina, infatti, all'inizio della formattazione, va a picchiare contro di esso, e stabilisce il punto di partenza della traccia numero uno.

Il continuo battere ad ogni errore, le sollecitazioni di lettura, il calore che caratterizza l'operato del nostro drive (che ha uno dispersore di calore pressoche' inesistente) hanno un effetto deleterio nei confronti del fissaggio di questo supporto, che risulta

dopo un po' di tempo spostato rispetto alla posizione originale.
Le conseguenze si vedono presto, o meglio si sentono: il rumore tende a subire una notevole variazione di tono e di intensita'. Durante la formattazione, il tempo che intercorre tra un "battito" e l'altro e' maggiore, e questo intuitivamente indica che il fermo e' piu' lontano del solito, e di conseguenza la testina impiega piu' tempo a raggiungerlo.

--------------------- listato n.25 - test n.1 --------------------

Fortunatamente l'assistenza tecnica in questi ultimi tempi e' migliorata e permette all'utente di aggiustare la propria macchina entro un ragionevole lasso di tempo. Fino a qualche mese fa erano talmente pochi i centri di assistenza che i tempi di attesa si aggiravano sui sei mesi per una riparazione di questo tipo.
Ma potremmo riparare noi il nostro 1541? Sono abbastanza scettico, anche perche' per quanto possiamo essere in gamba, questo tipo di intervento presuppone una preparazione tecnica non indifferente, e soprattutto una serie di accessori e di strumenti quali lo spessimetro di precisione, l'oscilloscopio, ed altre apparecchiature di misura per il segnale che noi poveri mortali, semplici appassionati, non possiamo possedere.
Nonostante cio' ritengo giusto imparare a trattare con una certa confidenza il nostro drive, soprattutto per sapere cosa non va e quale rimedio sarebbe opportuno applicare, per non prendere "fregature" e per sapere di quel che si parla quando si espone ad un tecnico il nostro problema.
L'inconveniente principale di una testina non correttamente allineata sono i difetti di formattazione e di lettura di dischi formattati da un altro drive. Spostatosi il fermo, la testina comincera' la formattazione in una zona magnetica non usata dagli altri lettori; ecco nascere l'incompatibilita' in lettura, per cui se faremo leggere al nostro 1541 un supporto formattato correttamente, in traccia uno riceveremo un errore come se li' non fosse giunta la testina per l'inizializzazione del dischetto.
Le traccie interessate sono quelle estreme, la prima o la 35esima, ma in caso di un'allontanamento maggiore non e' escluso che il disavanzo possa essere maggiore.
Ma proviamo a smontare il nostro drive, con pazienza ed una certa delicatezza, seguendo le seguenti istruzioni:

1 - Ruotate di 180 gradi l'apparecchio, in modo tale da vedere le quattro viti che mantengono chiusa la scatola di contenimento.
2 - Svitiamo le quattro viti non usando cacciaviti magnetici, e cercando di operare in modo simmetricamente obliquo (come se togliessimo i bulloni dalla ruota della nostra auto).
3 - Estratte le viti (assolutamente da non perdere), ruotiamo nuovamente di 180 gradi la scatola riportando in posizione normale il drive.
4 - Svitare le sei viti che tengono in un pezzo unico la parte meccanica al bordo.
5 - Rimuovete le due viti del coperchio di protezione di metallo e togliete il coperchio, staccando i fili che collegano il LED alla meccanica.
6 - Collegate con molta precauzione il cavo seriale al computer ed

il cavo di alimentazione nella presa posteriore apposita.

Penso sia inutile ricordarvi che state agendo su un apparecchio a 220 Volt, che puo' uccidervi tranquillamente nel caso tocchiate qualcosa che non va assolutamente toccato (le raccomandazioni non sono mai troppe).

Una volta aperto l'apparecchio, ci troviamo davanti la panoramica della macchina; possiamo idealmente dividerla in due parti. Una parte meccanica, comprendente il motorino, la testina di lettura, il tampone circolare, e la struttura metallica di sostegno. Una parte magnetica, dove campeggiano le ROM e le RAM del nostro drive, e che noi non toccheremo assolutamente, in quanto i contatti di questo tipo sono delicatissimi, ed una saldatura a stagno e' molto facile da rovinare.

--------------------- listato n.27 - test n.2 ---------------------

Osservate la testina, in ceramica, caratterizzata da una riga nera centrale. Essa va pulita periodicamente, o con dei kit o tramite l'uso di cotone idrofilo e di alcool etilico puro, per non laciare residui sulla sua superficie. Toccarla con le dita vuol dire rovinarla in modo quasi irreversibile, poiche' il grasso delle nostre dita coprirebbe la sua superficie impedendo una corretta lettura magnetica dei dati.

I listati presentati in questo capitolo hanno una funzione puramente dimostrativa, e secondo me non possono essere giudicati adatti ad una riparazione efficace di un drive disallineato. Tramite loro pero' possiamo vedere come si comporta il motorino, come si muove la testina, dietro quali impulsi si accende il LED, ecc.

I listati

Il primo programma e' basato su quattro controlli principali:

1 - test sull'efficacia della protezione da scrittura: occorre inserire un dischetto formattato e protetto da una linguetta adesiva, ed il programma provera' a scriverci sopra. Se il LED lampeggia, a segnalare un errore, il test e' stato superato.

2 - test sull'allineamento della testina: la testina va a leggere in traccie a caso dalla numero 35 alla numero 1, e questo per cinque volte. Se non vi sono errori di lettura, sintomo di errata formattazione del supporto, il test e' superato.

3 - test di compatibilita' di due drive, collegati tra loro. Viene effettuata la lettura e la scrittura di due dichetti, che vengono poi comparati. Come test non e' molto efficace, perche' nato per un sistema superiore direttamente a due drives.

4 - test sulla funzionalita' del LED luminoso, che ad un preciso comando deve lampeggiare.

Questo primo listato puo' essere fatto funzionare a drive chiuso.

Il secondo programma, invece, richiederebbe l'apertura del drive

per il controllo del movimento della puleggia. In piu' e' possibile calcolare la calibratura tra lo stop ed il passaggio della testina.

------------------- listato n.28 - test drive 3 ------------------

Il terzo test, forse il piu' efficace, ci da' la possibilita' di sbizzarrirci scorrendo sul dischetto come su di una pista di pattinaggio, di saltare da un blocco all'altro, di effettuare test di lettura e di scrittura sul supporto.
Ma ecco le istruzioni:

I - Sposta la testina sul blocco successivo; una routine controlla che non si cerchi di accedere a indirizzi illegali.

O - Sposta la testina sul blocco precedente.

B - In qualsiasi punto si trovi la testina, viene riportata al centro del dischetto, in traccia 17.

H - La testina effettua dei salti casuali dauna parte all'altra del supporto.

D - Fine del programma.

T - Il programma effettua un test di scrittura e di lettura sul disco, che deve essere formattato e privo dell'etichetta di protezione.

Viene anche effettuato un test sulla distanza tra il fermo ed il movimento della testina. In questo caso il drive dovrebbe essere aperto,manon e' poi cosi' indispensabile.

Insomma, se volete un consiglio, controllate pure se il vostro drive risulta disallineato, ma non fate nulla per tentare di aggiustarlo da soli. Meglio spendere un po' di tempo e di denaro piuttosto che rovinare l'apparecchio o correre il rischio di morire folgorati.

Capitolo 11: Disk drive 1571

Il drive 1541 di cui abbiamo gia' ampiamente parlato presenta due limitazioni fondamentali: la velocita' e la capacita'. Per queste ragioni il suo impiego e' ristretto ad un uso principalmente hobbistico, con gran dispiacere di tutti coloro che utilizzano i computer Commodore per scopi professionali oltreche' per giocarci. Per queste ragioni e per venire incontro alle esigenze di tutti coloro che volevano rendere il loro sistema piu' completo e professionale la Commodore ha messo sul mercato dapprima il drive 1570 e piu' recentemente il 1571, quest' ultimo particolarmente adatto a lavorare in stretta simbiosi con il computer Commodore 128.

Anzitutto anche il 1571 e' un drive intelligente, cioe' fornito di un microprocessore interno (e precisamente di un 6502A) che si occupa di comunicare con l'unita' centrale tramite il bus seriale e di gestire fisicamente il disco. Inoltre questo drive e' in grado di fare molte cose interessanti come ad esempio simulare il funzionamento e tutte le caratteristiche interne di un 1541 o leggere dischi formattati con standard completamente diversi da quelli della Commodore.

Ma andiamo per gradi. Cominciamo anzitutto a vedere quelle che sono le sue caratteristiche fondamentali:

- Doppia testina (cioe' doppia faccia)
- Capacita' totale formattata: 349696 bytes
- Massima dimensione dei file sequenziali: 337312 bytes
- Massima dimensione dei file relativi: 161132 bytes
- Massimo numero di files distinti per disco: 144
- Record per file: 65535
- Traccie per disco: 70
- Settori per ogni traccia: da 17 a 21
- Numero di bytes per settore: 256
- Numero di settori per disco: 1366 (totali), 1328 (disponibili)
- 32 Kb di ROM interna
- 2 Kb di RAM interna

Dalle caratteristiche si vede subito anzitutto che la capacita' e' doppia rispetto al vecchio 1541, difatti ora abbiamo due testine distinte (una per lato), che permettono di utilizzare dischi doppia faccia senza essere costretti a ritagliarli ma soprattutto senza doverli girare da un lato all'altro.

Parliamo ora un pochino della velocita'. Il drive 1571 e' abilitato al caricamento dei dati e dei programmi con una velocita' che puo' essere da cinque a sei volte superiore a quella del 1541 tuttavia questa caratteristica puo' essere sfruttata solo con il Commodore 128 che e' fornito di una gestione veloce dei dati sul bus seriale. Mentre se usato con il CBM 64 che tratta i dati sul bus seriale alla solita velocita' questa caratteristica non puo' essere sfruttata. Rimangono tuttavia i notevoli vantaggi della doppia capacita' e altri che vedremo in seguito.

Come gia' accennato precedentemente il 1571 puo' simulare il vecchio 1541 e potremo cosi' avere due modi operativi principali

differenti:

MODO OPERATIVO 1571

E' il modo operativo principale che entra automaticamente in funzionamento quando il drive e' collegato ad un Commodore 128 sin dalla sua accensione. Questo modo operativo consente il trasferimento veloce dei dati e la doppia capacita'. Puo' anche essere ottenuto dopo l'accensione tramite il comando del basic:

```
OPEN 15,8,15,"U0>M1":CLOSE15
```

questo comando puo' essere utilizzato sia con il Commodore 128 che con il CBM 64 anche se, come gia' spiegato, in quest'ultimo caso potremo si' utilizzare la doppia capacita' ma la velocita' di caricamento restera' quella usuale del 1541. Centriamo ora un attimo l'attenzione sulla precedente istruzione, in essa compare il numero 8 che rappresenta il numero di device fondamentale dell'unita' drive.Questo numero e' molto importante quando si lavora con piu' di un drive e puo' essere selezionato sui valori 8, 9, 10 o 11 tramite un piccolo commutatore presente sulla parte posteriore del drive. Qalora sia stato precedentemente cambiato bisognera' ovviamente tenerne conto nell'impartire l'istruzione. Ultima ma importantissima caratteristica di questo modo operativo e' la capacita' di leggere e scrivere su dischi formattati in uno standard (MFM modified frequency modulation) differente da quello della Commodore (chiamato GCR group code recording) e questa peculiarita' sara' di fondamentale importanza quando si lavorera' in ambiente CP/M.

MODO OPERATIVO 1541

All'atto dell'accensione il drive entra automaticamente in funzionamento in questo modo operativo (ricordiamo qui' inoltre che i drive e tutte le periferiche in generale devono sempre essere accesi prima dell'unita' centrale). Quando poi il computer verra' acceso, se e' un Commodore 128 provvedera' autamaticamente a settare il drive in modo operativo 1571, mentre se e' un CBM 64 rimarra' inserito il modo operativo 1541. Dato che del drive 1541 abbiamo gia' ampiamente parlato ci limitiamo qui' a ricordate che e' un singola faccia abilitato a leggere i dischi formattati esclusivamente in GCR con solo 35 traccie. In questo modo operativo la simulazione del drive 1541 e' totale cioe' compatibile al 100 % e potranno quindi essere utilizzati i programmi tipo Turbo-Disk o cartuccie analoghe. Questo modo operativo puo' comunque essere ottenuto dopo l'accensione del computer con il comando basic:

```
OPEN 15,8,15,"U0>M0":CLOSE15
```

per cui valgono considerazioni analoghe all'altra istruzione riguardanti il numero di device. A beneficio degli utenti del CP/M ricordiamo che in questo modo operativo potranno essere caricati e salvati programmi in CP/M posti su dichi forattati in modo GCR

realizzati con la vecchia cartuccia CP/M del CBM 64.

I DISCHI

Per quanto riguarda i dischi e per qualunque dei due modi operativi si scelga e' bene comuque utilizzare dischetti doppia faccia doppia densita' (contraddistinti dalla sigla DS,DD), che saranno indispensabili per il funzionamento in modo operativo 1571, e potranno comunque essere utilizzati da entrambi i lati anche nell'altro modo operativo semplicemente realizzando un seconda tacca con delle forbici (o con appositi marghingegni in vendita nei computer-shop) e girandoli sull'altro lato. In quest'ultimo caso pero' seppure la capacita' viene raddoppiata non potremo accedere contemporaneamente ad entrambe i lati del disco.

Quando siamo in modo operativo 1571 possiamo lavorare su dischi che siano stati formattati in entramdi i modi operativi mantenendo le caratteristiche di velocita' tipiche del 1571. Tuttavia i dischi formattati dal 1541 resteranno comunque dei singola faccia ed avranno quindi una capacita' dimezzata, a meno di non riformattarli con conseguente perdita di tutti i dati immagazzinati.

Se invece siamo in modo operativo 1541 potremo ovviamente leggere senza alcun problema i dischi formattati dal 1541 mentre quelli formattati dal modo operativo 1571 potranno essere letti solo sul lato A (cioe' quello in cui risiede la directory). Non ci si faccia trarre in inganno dal fatto che la directory contiene ancora tutti i file (immagazzinati nell'altro modo operativo) perche' potremo comunque caricare solo quelli che risiedono interamente sul lato A. Gli altri file non sono affatto persi, bastera' tornare nell'altro modo operativo per poterli ricaricare.

INSTALLAZIONE E MANUTENZIONE

Installare un drive 1571 e' un'operazione semplicissima. Ma procediamo per passi:

Passo 1 - Estrarre il drive dalla sua scatola di imballagio (conservare quest'ultima nell'eventualita' di un sucessivo trasporto).

Passo 2 - Verificare che nella scatola siano inoltre effettivamente contenuti tutti gli oggetti elencati esternamente ad essa.

Passo 3 - Accertarsi che la tensione di alimentazione prevista sia 220V e che l'interruttore di accensione sia in posizione OFF quindi collegare il cavo di alimentazione ad una presa di rete e sull'altro capo alla presa a tre poli presente sul lato posteriore del drive.

Passo 4 - Collocarlo in una posizione che consenta il circolo dell'aria onde evitare eccessivi surriscaldamenti.

Passo 5 - Accertarsi che il computer sia spento e collegare quindi il cavo seriale alle prese SERIAL PORT sul retro del computer e del drive.

Passo 6 - Rimuovere il cartoncino inserito nella fessura del drive

(e conservarlo per eventuali sucessivi trasporti).

Passo 7 - Accertarsi che il commutatore di numero device sia posto sull'otto e infine accendere prima il drive e poi il computer.
Come ultima raccomandazione trattate sempre bene i dischi e provvedete almeno un volta l'anno (o piu' di frequente per un uso intensivo) a far pulire e lubrificare le parti interne del drive presso un centro tecnico Commodore.

COMANDI BASIC

Il drive 1571 essendo compatibile sia con il CBM 64 che con il Commodore 128 e' in grado di accettare e quindi eseguire i comandi del basic 2.0 e del basic 7.0 che sono i linguaggi installati nei due computer sopra menzionati. Inoltre tali comandi li accettera' in entambi i modi operativi del drive.

Per quanto riguarda i comandi del basic 2.0 (cioe' del CBM 64) valgono tutte le considerazioni esposte nella prima parte del libro. L'unica aggiunta e' ovviamente il comando che permette al drive di settare uno dei due modi operativi permessi e di cui abbiamo gia' ampiamente parlato.

Il basic 7.0 e' un linguaggio molto piu' evoluto che, tra le altre cose, contiene al suo interno anche tutti i comandi del basic 2.0 , cio' consentira' inoltre di trasportare sul Commodore 128 e quindi sul drive 1571 tutti i programmi basic 2.0 sviluppati sul CBM 64 (e drive 1541) consentendo inoltre di guadagnare in velocita' e capacita.

- I COMANDI DEL BASIC 7.0

Un primo grande vantaggio del basic 7.0 e' il nuovo metodo per la rilevazione degli errori. Infatti quando vedremo lampeggiare furiosamente il led rosso potremo avere immediatamente un'idea dello stato del disco perche' questo sara' memorizzato nella variabile DS$ (che e' una parola riservata del basic) e con una semplice istruzione PRINT DS$ otterremo il messaggio di errore relativo al drive. Elenchiamo ora i comandi del basic 7.0. Si tenga presente che tutte le lettere maiuscole andranno riportate tali e quali nei comandi, mentre quelle minuscole indicheranno dei paramtri che andranno aggiunti dall'operatore; quando infine un parametro sara' racchiuso da parentesi quadre [...] cio' indichera' che tale parametro e' opzionale e se omesso verra' assunto quello di default (cioe' quello tipico all'atto dell'accensione). Valgono inoltre tutte le considerazioni gia' fatte nella prima parte del libro per quanto riguarda le dimensioni dei nomi dei file e l'uso dei due caratteri * e ? per sostituire parti mancanti di parole o caratteri mancanti. Ultima ma importante considerazione e' che quando un parametro stringa (come ad esempio nomefile) viene sostituito da una variabile alfanumerica (ad esempio A$) questa dovra' essere racchiusa tra parentesi (A$) anziche' tra virgolette. Ad esempio se si pone A$="nomefile" le due istruzioni DSAVE"nomefile" e DSAVE(A$) sono equivalenti.

-DSAVE"nomefile"[,Udevice][,Ddrive]

Questa istruzione permette il salvataggio di un programma su disco, nomefile e' il nome che si assegna al file (programma in questione) ed e' obbligatorio, se inoltre il nome del file e' peceduto dal carattere @ allora questo file andra' a sostituirne uno vecchio con lo stesso nome; device e' il numero di device di cui abbiamo gia' parlato e che puo' essere 8, 9, 10 o 11 (se omesso viene assunto automaticamente 8), infine drive e' un parametro che interessa solo i drive doppi e puo' essere 0 o 1.

-BSAVE"nomefile"[,Udevice][,Ddrive][,Bbank][,Pindiniz TO indfin+1]

Con questa istruzione possiamo salvare zone di memoria comprese tra indiniz che e' l'indirizzo iniziale e indfin che e' l'indirizzo finale (nel comando questo valore andra' aumentato di uno). I parametri device e drive sono identici a quelli dell'istruzione DSAVE. Il parametro bank dovra' essere compreso tra 0 e 15 e indichera' uno dei sedici banchi di memoria disponibili sul Commodore 128 e che indichiamo appresso:

```
   banco 0: solo RAM(0)
   banco 1: solo RAM(1)
 * banco 2: solo RAM(2)
 * banco 3: solo RAM(3)
   banco 4: ROM interna, RAM(0), Input/Output
   banco 5: ROM interna, RAM(1), Input/Output
 * banco 6: ROM interna, RAM(2), Input/Output
 * banco 7: ROM interna, RAM(3), Input/Output
   banco 8: ROM esterna, RAM(0), Input/Output
   banco 9: ROM esterna, RAM(1), Input/Output
 * banco10: ROM esterna, RAM(2), Input/Output
 * banco11: ROM esterna, RAM(3), Input/Output
   banco12: KERNAL e ROM(low) interna, RAM(0), Input/Output
   banco13: KERNAL e ROM(low) esterna, RAM(0), Input/Output
   banco14: KERNAL e ROM(basic), RAM(0), ROM(caratteri)
   banco15: KERNAL e ROM(basic), RAM(0), Input/Output
```

(I banchi segnati con l'asterisco sono disponibili solo congiuntamente ad espansioni di RAM esterne)

-DVERIFY"nomefile"[,Udevice][,Ddrive]

Questa e' una semplice istruzione che va a verificare che il programma in memeoria sia identico a quello salvato sul disco con il nome nomefile. I parametri device e drive sono identici a quelli dell'istruzione DSAVE.

-COPY[Ddrive,]"nomefilev"TO"[Ddrive,]"nomefilen"[,Udevice]

Con questo comando possiamo copiare un file che abbia come nome nomefilev in un nuovo file che verra' chiamato nomefilen. Device e' un parametro identico a quello usato nell'istruzione DSAVE. La copia deve essere effettuata sullo stesso disco oppure se si dispone di

un'unita' drive doppia puo' essere effettuata dal drive Ø al drive 1 o viceversa specificandolo nell'istruzione con i due parametri drive.

-CONCAT[Ddrive,]"nomefile1"TO[Ddrive,]"nomefile"[,Udevice]

Questa istruzione permette di aggiungere il file chiamato nomefile1 al file chiamato nomefile. Entrambi i file devono essere sequenziali. Se si dispone di un'unita' drive doppia e' possibile che i due file siano posti su due dischi diversi e cio' andra' ovviamente specificato con i due parametri drive. Il parametro device e' analogo a quello dell'istruzione DSAVE.

-SCRATCH"nomefile"[,Udevice][,Ddrive]

Quando un file non e' piu' necessario puo' essere eliminato dal disco con questo comando che lo cancella completamente. I parametri device e drive sono quelli usuali. Il computer rispondera' con la domanda -ARE YOU SURE ?- per farvi capire che state andando a cancellare irrimediabilmente un file, se difatti volete cancellarlo rispondete con Y (che sta per YES cioe' si).

-RENAME[Ddrive,]"nomefilev"TO[Ddrive,]"nomefilen"[,Udevice]

A volte si puo' rendere necessario cambiare il nome di un file per svariati motivi e cio' puo' essere ottenuto con questa istruzione dove nomefilen e' il nuovo nome che verra' assegnato al file nomefilev. I parametri device e drive sono quelli usuali

-COLLECT

Questa istruzione permette analogamente all'istruzione VALIDATE del basic 2.Ø di di riordinare la BAM (block availability map) liberando il maggior numero di blocchi disponibili che per qualche ragione erano stati precedentemente occupati e cancellati malamente.

-BACKUP Ddriveo TO Ddrivec [,Udevice]

Quando e solo quando si dispone di un'unita' drive doppia (e non di due drive separati) puo' essere utilizzato questo comando per realizzare una copia perfetta di un disco. Il parametro driveo deve essere Ø o 1 ed indica il drive in cui va inserito il disco sorgente, mentre il parametro drivec deve essere 1 o Ø ed indica il drive in cui va inserito un disco vuoto su cui realizzare la copia, questo disco non necessita di essere precedentemente formattato ed eventuali dati presenti su di esso prima di questo comando andranno irrimediabilmente perduti. Per questo motivo il computer risponde a questo comando con la domanda ARE YOU SURE? per farvi capire che state per modificare irrimediabilmente qualcosa; se volete che il comando abbia esito dovrete ovviamente rispondere Y (che significa si).

-DLOAD"nomefile"[,Udevice][,Ddrive]

Un programma che sia stato precedentemente salvato su disco puo'

essere ricaricato con questo comando, nomefile indica il nome del programma che si vuole ricaricare, mentre i parametri device e drive sono quelli usuali indicanti il numero della periferica (8, 9, 10 o 11) e del drive (solo per unita' drive doppie, 0 o 1).

-BLOAD"nomefile"[,Udevice][,Ddrive][,Bbank][,Pindiniz]

Questo comando permette di ricaricare una zona di memoria che sia stata precedentemente salvata con il comando BSAVE. I parametri device, drive, e bank sono identici a quelli dell'istruzione BSAVE. Il parametro indiniz, se messo indica l'indirizzo di memoria di partenza in cui si vuole che avvenga il caricamento se invece tale parametro viene omesso il file sara' ricaricato nella stessa zona di memoria da cui e' stato salvato.

-BOOT["nomefile"][,Udevice][,Ddrive]

I parametri di questo comando sono tutti opzionali. Il suo effetto e' quello di caricare un file in linguaggio macchina e farne automaticamente partire l'esecuzione. Se il nome del file viene omesso verra' caricato ed eseguito il primo file in liguaggio macchina che verra' incontrato sul disco (cioe' un file di tipo USR). I perametri device e drive sono quelli usuali. Tale comando e' ad esempio indispensabile per attivare il Commodore 128 al modo di funzionamento in CP/M con il microprocessore Z80 interno.

-DIRECTORY[Udevice][,Ddrive][,"nomefile"]

-CATALOG[Udevice][,Ddrive][,"nomefile"]

Questi due comandi sono perfettamente identici e permettono di visualizzare sullo schermo la directory del disco che si trova nel drive specificato dai parametri device e drive, senza andare a modificare nulla ne in memoria ne sul disco. Il nomefile sara' in generale una stringa in cui compariranno i caratteri jolly * e ? che permetteranno di avere la directory solo di alcuni file. Ad esempio con il comando DIRECTORY"a*" avremo la directory del drive 0 sull'unita' periferica 8 di tutti i file il cui nome inizia con la lettera a.

-DCLOSE[#nfilogico][,Udevice]

Questo comando permette di chiudere un file aperto sul disco posto nel drive individuato dal numero di device(8, 9, 10 o 11). Se nfilogico che e' il numero del file logico con cui quel file e' stato aperto e' omesso verranno chiusi tutti i file aperti su quel disco.

-DOPEN#nfilogico,"nomefile[,S/P]"[,Lreclung][,Udevice][,Udrive][,w]

Con questo comando possiamo aprire un file chiamato nomefile sul disco per leggerci e/o scriverci dentro. Il parametro nfilogic e' il valore che noi assegnamo al file logico e deve essere un valore compreso tra 0 e 255 (valori maggiori di 128 compreso forzano un RETURN ad ogni scrittura PRINT# nel file). Alla fine del nomefile puo

essere messa (separata da una virgola) la lettera S che indica file di tipo sequenziale oppure la lettera P che indica file di tipo programma. Il parametro reclung rappresenta la lunghezza del record (solo per i file relativi). I parametri device e drive sono quelli usuali. Infine la lettera w se messa indica un file aperto in scrittura, se omessa un file aperto in lettura (i file relativi ovviamente non necessitano questo parametro dato che sono aperti contemporaneamente in lettura e in scrittura).

-HEADER"nomedisco"[,Iid][,Udevice][,Ddrive]

La formattazione dei dischi puo' essere effettuata con questo comando. A seconda del modo operativo in cui ci si trova cioe' 1541 o 1571 il disco sara' formattato in singola faccia o doppia faccia. Il parametro nomedisco e' il nome che assegnamo al disco e il parametro id e' l'identificatore del disco (che puo' essere omesso). I due parametri device e drive sono quelli usuali. A questo comando il computer risponde con la domanda ARE YOU SURE? per farvi capire che state compiendo qualcosa di irreparabile; se volete che il disco sia formattato dovete ovviamente rispondere Y (per si).

-RECORD#nfilogic,numrecord[,byte]

Con questa intruzione posizioniamo il puntatore al record di un file relativo con il parametro numrecord. Eventualmente possiamo posizionarci all'interno del record in un punto qualsiasi con il parametro byte. Il parametro nfilogic rappresenta il numero di file logico stabilito all'atto dell'apertura del file con l'istruzione DOPEN.

```
100 REM ********************************
105 REM            LISTATO N 1
106 REM ********************************
110 REM             TEST DISCO
120 REM ********************************
130 DN=8:NM$="DISCO TEST 001":ND$="99"
140 DIMT(100):DIMS(100)
150 POKE53280,0:POKE53281,7:POKE646,0
160 PRINT"█────────────────────────────────────────"
170 PRINT"██ TEST LETTURA E SCRITTURA SU DISCO 1.1  █"
180 PRINT"█────────────────────────────────────────"
190 IFDF=0THENGOTO580
200 D$="0"
210 OPEN15,DN,15
220 PRINT#15,"V"D$
230 N%=RND(TI)*255
240 A$="":FORI=1TO255:A$=A$+CHR$(255AND(I+N%)):NEXT
250 GOSUB540
260 OPEN2,DN,2,"#"
270 PRINT:PRINT#2,A$;
280 T=1:S=0
290 PRINT#15,"B-A:"D$;T;S
300 INPUT#15,EN,EM$,ET,ES
310 IFEN=0THEN340
320 IFET=0THEN420:REM END
330 PRINT#15,"B-A:"D$;ET;ES:T=ET:S=ES
340 PRINT#15,"U2:2,"D$;T;S
350 NB=NB+1:PRINT"████  BLOCCHI TESTATI N."NB
360 PRINT"█  █ TRACCIA    █████"T;" █ SETTORE     █████"S"█████"
370 INPUT#15,EN,EM$,ES,ET
380 IF EN=0THEN280
390 T(J)=T:S(J)=S:J=J+1
400 PRINT"██████████████  █ BLOCCO ROVINATO:███",T;S"██████"
410 GOTO280
420 PRINT#15,"I"D$
430 GOSUB540
440 CLOSE2
450 IFJ=0THENPRINT"███████  TEST NEGATIVO: OK!":END
460 OPEN2,DN,2,"#"
470 PRINT"███████ BLOCCHI ROV.","█ TRACCIA","█ SETTORE"
480 FORI=0TOJ-1
490 PRINT#15,"B-A:";D$,T(I);S(I)
500 PRINT,,T(I),S(I)
510 NEXT
520 PRINT"█"J" BLOCCHI ROVINATI SONO PRESENTI
530 CLOSE2:END
540 INPUT#15,EN,EM$,ET,ES
550 IF EN=0 THEN RETURN
560 PRINT"██ERRORE #"EN,EM$;ET;ES""
570 PRINT#15,"I"D$
580 REM ********************************
590 REM           FORMATTAZIONE
600 REM ********************************
610 PRINT"████ IL DISCO E' GIA' FORMATTATO ? (S/N)
```

```
620 GETSI$:IFSI$=""THEN620
630 IFSI$="S"THENDF=1:GOTO160
640 IFSI$<>"N"THEN620
650 PRINT"██ STO FORMATTANDO IL DISCO "
660 PRINT"█ NOME : █"NM$;"█ ID : █";ND$
670 OPEN1,8,15,"N0:"+NM$+","+ND$
680 CLOSE1
690 PRINT"█ FORMATTAZIONE ESEGUITA "
700 FORTY=0TO1000:NEXTTY
710 PRINT" INIZIO IL TEST "
720 FORTY=0TO1000:NEXTTY
730 GOTO160
```

```
100 REM ********************************
110 REM           LISTATO N 2
115 REM ********************************
120 REM   UTILITY DISK PER C64 & 1541
160 REM ********************************
170 REM INIZIALIZZAZIONE
180 POKE53280,0:POKE53281,0:POKE646,7
190 MX=144:DIM SC$(MX)
200 OPEN15,8,15
210 REM LOOP PRINCIPALE
220 GOSUB 1140 :REM STAMPA MENU
230 GETT$:IFT$<CHR$(133)ORT$>CHR$(140)      THEN 230
240 PRINT"█ F";:A=ASC(T$)
250 IF A<137THENGOTO270
260 PRINTCHR$((A-137)*2+50);" █":GOTO280
270 PRINTCHR$((A-133)*2+49);" █"
280 FORT=0TO500:NEXT:PRINTCHR$(147);
290 ON ASC(T$)-132 GOSUB1310,460,930,      820,520,1740,1040,780
300 GOTO 220 :REM FINE DEL LOOP
310 REM SUBROUT. DI ERRORE
320 INPUT#15,A$,B$,C$,D$
330 IF VAL(A$)<20 THEN RETURN
340 PRINT:PRINT"██              DISK ERROR                ██"
350 PRINT
360 PRINT"ERRORE NUMERO ";A$;
370 PRINT" IN TRACCIA ";C$;" ,SETTORE ";D$
380 PRINT"DESCRIZIONE ERRORE : ";B$
390 PRINT:PRINT"MI SPIACE..."
400 GETT$:IFT$<>CHR$(13)THEN400
410 RETURN
420 REM RITARDO PER IL DISCO (SUB)
430 PRINT"INSERISCI IL DISCO"
440 GET T$:IF T$<>CHR$(13)THEN440
450 RETURN
460 REM VALIDATE DEL DISCO
```

```
470 GOSUB 420 :REM INSERIMENTO DISCO
480 PRINT:PRINT"ATTENDERE"
490 PRINT#15,"VALIDATE"
500 GOSUB 310 :REM TESTA PER ERRORI I/O
510 RETURN
520 REM FORMATTA UN DISCO
530 GOSUB 650 :REM NOME DEL DISCO
540 GOSUB 600 :REM IDENT. DEL DISCO
550 GOSUB 420 :REM ATTESA
560 PRINT:PRINT"ATTENDERE"
570 PRINT#15,"N0:";NA$;",";ID$
580 GOSUB 310 :REM TESTA PER ERRORI I/O
590 RETURN
600 REM ID DEL DISCO
610 ID$="":INPUT"ID ";ID$
620 IF LEN(ID$)=2 THEN RETURN
630 PRINT"L'ID DEVE ESSERE DI 2 CARATTERI"
640 PRINT:GOTO 610
650 REM NOME DEL DISCO
660 NA$="":INPUT"NOME DEL DISCO ";NA$
670 IFNA$<>""ANDLEN(NA$)<=18THENRETURN
680 PRINT"IL NOME DEVE ESSERE UNA STRINGA DI NON"
690 PRINT"PIU' DI 18 CARATTERI"
700 PRINT:GOTO 660
710 REM SCRIVE HD$ E DA' IL NOME AL FILE
720 PRINT HD$;
730 NA$="":INPUT NA$
740 IFNA$<>""ANDLEN(NA$)<=16THENRETURN
750 PRINT"IL NOME DEL FILE E' UNA STRINGA NON "
760 PRINT"NULLA DI 16 CARATTERI AL MASSIMO"
770 PRINT:GOTO 720
780 REM USCITA
790 PRINT CHR$(147);"TANTI SALUTI":PRINT
800 CLOSE15
810 END
820 REM LOAD & RUN UN PROGRAMMA
830 CLR:HD$="NOME DEL PRG "
840 GOSUB 710 :REM NOME DEL FILE
850 OPEN15,8,15:OPEN2,8,2,NA$
860 GOSUB 310 :REM TESTA PER ERRORI I/O
870 IF VAL(A$)<>0THEN CLOSE2:RUN
880 CLOSE2:PRINT:PRINT"LOADING ";NA$
890 CLOSE15:LOAD NA$,8,1
900 OPEN15,8,15
910 GOSUB 310 :REM TESTA PER ERRORI I/O
920 CLOSE15:RUN
930 REM FA LA COPIA DI UN FILE
940 HD$="VECCHIO FILE "
950 GOSUB 710 :REM NOME DEL FILE
960 OLD$=NA$
970 HD$="NUOVO FILE "
980 GOSUB 710 :REM NOME DEL FILE
990 GOSUB 420 :REM ATTENDE IL DISCO
1000 PRINT:PRINT"COPIO ";OLD$;" IN ";NA$
```

```
1010 PRINT#15,"C0:";NA$;"=0:";OLD$
1020 GOSUB 310 :REM TESTA PER ERRORI I/O
1030 RETURN
1040 REM CAMBIA IL NOME DI UN FILE
1050 HD$="VECCHIO NOME "
1060 GOSUB 710 :REM NOME DEL FILE
1070 OLD$=NA$
1080 HD$="NUOVO NOME "
1090 GOSUB 710 :REM NOME DEL FILE
1100 GOSUB 420 :REM ATTENDE IL DISCO
1110 PRINT#15,"R0:";NA$;"=";OLD$
1120 GOSUB 310 :REM TESTA PER ERRORI I/O
1130 RETURN
1140 REM SCRIVE IL TESTO MENU' PRINCIPALE
1150 PRINT CHR$(147);
1160 PRINT"________________________________________"
1170 PRINT"□■■      UTILITY DISK PER C64 & 1541       ■■"
1180 PRINT"□────────────────────────────────────────"
1190 PRINT"□ (C) EDISOFT SRL MILANO TEL: 02-2893580"
1200 PRINT:PRINT"■ F1 ■__DIRECTORY_DI_UN_DISCO
1210 PRINT:PRINT"■ F2 ■__FORMATTA_UN_NUOVO_DISCO
1220 PRINT:PRINT"■ F3 ■__VALIDATE_DI_UN_DISCO
1230 PRINT:PRINT"■ F4 ■__CANCELLA_FILES_INDESIDERATI
1240 PRINT:PRINT"■ F5 ■__COPIA_DI_UN_FILE
1250 PRINT:PRINT"■ F6 ■__CAMBIA_IL_NOME_DI_UN_FILE"
1260 PRINT:PRINT"■ F7 ■__LOAD_&_RUN_UN_PROGRAMMA
1270 PRINT:PRINT"■ F8 ■__FINE"
1280 PRINT:PRINT
1290 PRINT"DIGITA LA TUA SCELTA: ";
1300 RETURN
1310 REM DIRECTORY
1320 GOSUB 420 :REM ATTENDE IL DISCO
1330 OPEN1,8,0,"$0"
1340 GOSUB 310 :REM TESTA PER ERRORI I/O
1350 IF VAL(A$)<>0THENCLOSE1:RETURN
1360 GET#1,A$,B$
1370 GET#1,A$,B$
1380 GET#1,A$,B$
1390 GET#1,B$
1400 IF B$<>CHR$(34) THEN GOTO 1390
1410 PRINTCHR$(147);"■NOME DISCO:■ ";
1420 GET#1,B$:IFB$<>CHR$(34)THENPRINTB$;:GOTO1420
1430 PRINT:PRINT
1440 GET#1,B$:IFB$=CHR$(32)THEN1440
1450 C$=""
1460 C$=C$+B$:GET#1,B$:IFB$<>""THEN1460
1470 PRINT"■IDENT. :■ ";LEFT$(C$,3);
1480 PRINTTAB(16)"■FORMAT :■ ";MID$(C$,4)
1490 PRINT
1500 PRINT"■#BLK  NOME FILE        TIPO■"
1510 GET#1,A$,B$
1520 GET#1,A$,B$
1530 C=0
1540 IF A$<>""THEN C=ASC(A$)
```

```
1550 IF B$<>""THEN C=C+ASC(B$)*256
1560 PRINTMID$(STR$(C),2);
1570 GET#1,B$:IF ST<>0 THEN GOTO 1700
1580 IF B$<>CHR$(34) THEN GOTO 1570
1590 PRINTTAB(5)CHR$(34);
1600 GET#1,B$:IFB$<>CHR$(34)THENPRINTB$;:GOTO1600
1610 PRINT CHR$(34);
1620 GET#1,B$:IFB$=CHR$(32)THEN1620
1630 C$="":PRINTTAB(24);
1640 C$=C$+B$:GET#1,B$:IFB$<>""THEN1640
1650 PRINTLEFT$(C$,3)
1660 GETT$:IFT$=""THENGOTO1690
1670 IFT$="Q"THENCLOSE1:RETURN
1680 GETT$:IFT$<>CHR$(13)THEN1670
1690 IF ST=0 THEN GOTO 1510
1700 PRINT" BLOCCHI LIBERI."
1710 CLOSE1
1720 GETT$:IFT$<>CHR$(13)THEN1720
1730 RETURN
1740 REM SCRATCH FILES
1750 GOSUB 420:REM ATTENDE DISCO
1760 OPEN1,8,0,"$0"
1770 GOSUB 310 :REM TESTA ERRORI I/OR
1780 IF VAL(A$)<>0THENCLOSE1:RETURN
1790 GET#1,A$,B$
1800 GET#1,A$,B$
1810 GET#1,A$,B$
1820 GET#1,B$
1830 IF B$<>CHR$(34) THEN GOTO 1820
1840 PRINTCHR$(147);"■NOME DISCO:■ ";
1850 GET#1,B$:IFB$<>CHR$(34)THENPRINTB$;:GOTO1850
1860 PRINT:PRINT
1870 GET#1,B$:IFB$=CHR$(32)THEN1870
1880 C$=""
1890 C$=C$+B$:GET#1,B$:IFB$<>""THEN1890
1900 PRINT
1910 IX=0
1920 PRINT"■■BLK  NOME FILE        TIPO■  ■SCRATCH■"
1930 GET#1,A$,B$
1940 GET#1,A$,B$
1950 C=0
1960 IF A$<>""THEN C=ASC(A$)
1970 IF B$<>""THEN C=C+ASC(B$)*256
1980 PRINTMID$(STR$(C),2);
1990 GET#1,B$:IF ST<>0 THEN GOTO 2130
2000 IF B$<>CHR$(34) THEN GOTO 1990
2010 A$=""
2020 GET#1,B$:IFB$<>CHR$(34)THENA$=A$+B$:GOTO2020
2030 PRINTTAB(5)CHR$(34);A$;CHR$(34);
2040 GET#1,B$:IFB$=CHR$(32)THEN2040
2050 C$=""
2060 C$=C$+B$:GET#1,B$:IFB$<>""THEN2060
2070 PRINTTAB(24)LEFT$(C$,3);"   SI'/NO"
2080 GETT$:IFT$=""THENGOTO2080
```

```
2090 IFT$="Q"THENGOTO2130
2100 IFT$<>"S"THENPRINTTAB(30)"NO     ":GOTO2120
2110 SC$(IX)=A$:IX=IX+1:PRINTTAB(30)"SI'    "
2120 IF ST=0 THEN GOTO 1930
2130 CLOSE1
2140 IF IX=0 THEN RETURN
2150 TT=IX
2160 PRINTCHR$(147);
2170 PRINT"FILES DA SCRETCHARE :"
2180 PRINT
2190 FOR IX=0 TO TT-1
2200 PRINTSC$(IX)
2210 NEXT IX
2220 PRINT:PRINT"TOTALE ";TT;" FILES"
2230 PRINT:PRINT"CONFERMI ? ";
2240 GETT$:IFT$=""THEN2240
2250 IFT$<>"S"THENRETURN
2260 PRINT"SI'"
2270 PRINT:PRINT"SEI SICURO ? ";
2280 GETT$:IFT$=""THEN2280
2290 IFT$<>"S"THEN RETURN
2300 PRINT"SI'"
2310 FOR IX=0 TO TT-1
2320 PRINT:PRINT"SCRATCHING ";SC$(IX)
2330 PRINT#15,"S0:";SC$(IX)
2340 NEXT IX
2350 GOSUB 310 :REM TESTA PER ERRORI I/O
2360 PRINT:PRINT"FINE SCRATCH."
2370 GETT$:IFT$<>CHR$(13)THEN2370
2380 RETURN
```

```
100 REM******************
110 REM  LISTATO N 3
120 REM******************
130 REM PULIZIA  DISCO
140 REM******************
150 PRINTCHR$(14)
160 POKE53280,0:POKE53281,0:PRINT""
170 PRINT"              ULIZIA    ISCO           "
180 PRINT"UALE DRIVE (0-1) ?  ";
190 GETDR$:IFDR$<>"0"ANDDR$<>"1"THEN 190
200 PRINTDR$:D=VAL(DR$)
210 DR$="I"+DR$:OPEN15,8,15,DR$
220 GOSUB620
230 OPEN1,8,3,"#":GOSUB630
240 PRINT "     REMI  S  PER CANCELLARE":PRINT" TRACCIA 18: BLOCCO 1"
250 Z$=CHR$(0)
260 T=18:S=1
270 PRINT#15,"U1:"3;D;T;S
280 PRINT#15,"B-P:"3;2+32*R
```

```
290 GET#1,A$:IFA$=""THENA$=Z$
300 IFASC(A$)<129THEN400
310 FORK=5+32*RTO20+32*R
320 PRINT#15,"B-P:"3;K
330 GET#1,A$:IFA$=""THENPRINT"█ █";:GOTO350
340 PRINTA$;
350 NEXT
360 PRINT"█← DA CANCELLARE? (S/N)█";
370 GETA$:IFA$<>"S"ANDA$<>"N"THEN370
380 PRINTA$
390 IFA$="S"THENGOSUB490
400 R=R+1:IFR<8THEN270
410 PRINT#15,"B-P:"3;0
420 GET#1,A$:IFA$=""THENA$=Z$
430 T=ASC(A$):IFT=0THENPRINT:PRINT"_INE DELLA DIRECTORY":GOTO530
440 PRINT#15,"B-P:"3;1
450 GET#1,A$:IFA$=""THENA$=Z$
460 S=ASC(A$):PRINT:PRINT"█ TRACCIA  ";:NU=T:GOSUB690:PRINTN$;
470 PRINTTAB(20);": BLOCCO  ";:NU=S:GOSUB690:PRINTN$;:PRINT"█"
480 R=0:GOTO270
490 PRINT#15,"B-P:"3;5+32*R-3
500 PRINT#1,CHR$(0);
510 PRINT#15,"U2:"3;D;T;S
520 GOSUB630:RETURN
530 CLOSE1
540 PRINT"XUOI LA VALIDATE DEL DISCO ? (S/N)
550 GETV$:IFV$="S"THEN580
560 IFV$=""THEN550
570 CLOSE15:END
580 PRINT:PRINT"█VALIDATE IN CORSO
590 PRINT#15,("V"+DR$)
600 GOSUB630:CLOSE15
610 END
620 REM ***********************
630 INPUT#15,ER,ER$,TR,SE
640 IFER=0THENRETURN
650 CLOSE15:CLOSE1
660 PRINT:PRINT"█ERRORE";
670 PRINTER;ER$;TR;SE
680 STOP
690 REM ***********************
700 N$=MID$(STR$(NU),2)
710 IFLEN(N$)<2THENN$="0"+N$
720 RETURN
```

```
100 REM ***********************
105 REM       LISTATO N 4
108 REM ***********************
110 REM ORDINAMENTO ALFABETICO
120 REM  DIRECTORY DISCO 1541
130 REM ***********************
160 POKE53280,0:POKE53281,0:PRINT"■"
170 PRINTCHR$(14)"■                                   ■"
180 PRINT"■NSERISCI IL DISCO NEL DRIVE; PREMI           ■ QUANDO SEI PRO
NTO.
190 GETZ$:IFZ$<>CHR$(13)GOTO190
200 REM ***********************
210 DN=8:D=0:C2=2:CC=15:T=18:S=0:BP=144
220 OPENCC,DN,CC,"I0":GOSUB1070
230 OPENC2,DN,C2,"#":GOSUB1070
240 REM ***********************
250 PRINT#CC,"U1:"C2;D;T;S:GOSUB1070
260 PRINT#CC,"B-P:"C2;BP
270 ID$="":N=20:GOSUB1100:ID$=Z$
280 PRINT"■ISCO CARICATO :  "ID$" "
290 REM ***********************
300 DIMS%(19)
310 DIMFT$(192)
320 DIMTS$(192)
330 DIMNF$(192)
340 DIMFE$(192)
350 DIMSK$(192)
360 DIMT$(4)
370 T$(1)="SEQ":T$(2)="PRG":T$(3)="USR":T$(4)="REL"
380 FORI=1TO30:Z0$=Z0$+CHR$(0):NEXT
390 REM ***********************
400 S=1
410 E=1
420 PRINT#CC,"U1:"C2;D;T;S:GOSUB1070:BP=1
430 PRINT#CC,"B-P:";C2;BP:GOSUB1140:S%(S)=A
440 PRINT#CC,"B-P:"C2;BP;:GOSUB1140:FT=A
450 IFFT=0ORFT=128THENBP=BP+31:GOTO540
460 IFFT<128THENPRINT"■ ■            ■: FILE NON CHIUSO":GOTO1060
470 FT$(E)=A$
480 N=2:GOSUB1110:TS$(E)=Z$
490 N=16:GOSUB1110:NF$(E)=Z$
500 N=11:GOSUB1110:FE$(E)=Z$
510 REM ***********************
520 PRINT,"     "NF$(E)" : "T$(FT-128)
530 BP=BP+2:E=E+1
540 IFBP<256GOTO440
550 IFS%(S)<>255THENS=S%(S):GOTO420
560 GOSUB1170
570 E=E-1:IFE=0GOTO810
580 PRINT"|RUPPO DI FILE NORMALI ? (S/N)"
590 GETZ$:FT$=Z$:IFZ$="S"GOTO610
600 IFZ$<>"N"GOTO590
610 REM ***********************
620 PRINT"■TO RIORDINANDO..."
630 FORI=1TOE:K=16
640 FORJ=1TO16:IFMID$(NF$(I),J,1)=CHR$(160)THENK=J-1:J=16
```

```
650 NEXT:SK$(I)=LEFT$(NF$(I),K)
660 IFFT$="Y"THENSK$(I)=FT$(I)+SK$(I)
670 NEXT:M=E
680 M=INT(M/2):IFM=0GOTO810
690 K=E-M:J=0
700 IFJ>KGOTO680
710 I=J
720 IFI<0GOTO740
730 IFSK$(I+M)<SK$(I)GOTO750
740 J=J+1:GOTO700
750 A$=FT$(I):FT$(I)=FT$(I+M):FT$(I+M)=A$
760 A$=TS$(I):TS$(I)=TS$(I+M):TS$(I+M)=A$
770 A$=NF$(I):NF$(I)=NF$(I+M):NF$(I+M)=A$
780 A$=FE$(I):FE$(I)=FE$(I+M):FE$(I+M)=A$
790 A$=SK$(I):SK$(I)=SK$(I+M):SK$(I+M)=A$
800 I=I-M:GOTO720
810 REM *********************
820 PRINT"█♥TO SCRIVENDO...:   "ID$" "
830 Z$=CHR$(0)+CHR$(255)
840 S=1:N=1
850 PRINT#CC,"U1:"C2;D;T;S:GOSUB1070
860 BP=2
870 IFN>EGOTO950
880 IFBP<256GOTO910
890 PRINT#CC,"U2:"C2;D;T;S:GOSUB1070
900 S=S%(S):GOTO850
910 PRINT#CC,"B-P:";C2;BP
920 PRINT,"      "NF$(N)" :  "T$(ASC(FT$(N))-128)
930 PRINT#C2,FT$(N);TS$(N);NF$(N);FE$(N);
940 BP=BP+32:N=N+1:GOTO870
950 IFBP>255GOTO990
960 PRINT#CC,"B-P:";C2;BP
970 PRINT#C2,Z0$;
980 BP=BP+32:GOTO950
990 BP=0:PRINT#CC,"B-P:";C2;BP
1000 PRINT#C2,Z$;
1010 PRINT#CC,"U2:"C2;D;T;S:GOSUB1070
1020 Z$=LEFT$(Z0$,2)
1030 S=S%(S):IFS=255GOTO1060
1040 PRINT#CC,"U1:"C2;D;T;S:GOSUB1070
1050 BP=2:GOTO960
1060 CLOSEC2:CLOSECC:END
1070 REM *********************
1080 INPUT#CC,EN,EM$,ET,ES:IFEN=0THENRETURN
1090 PRINT"██ ‾_┌_‾ ♥ㄥ ‾\♥┤‾ ██"EN;EM$;ET;ES:GOTO1060
1100 REM *********************
1110 Z$=""
1120 FORI=1TON:GOSUB1140
1130 Z$=Z$+A$:NEXT:RETURN
1140 REM *********************
1150 GET#C2,A$:IFA$=""THENA$=CHR$(0)
1160 A=ASC(A$):BP=BP+1:RETURN
1170 REM *********************
1180 PRINT"█┌OSSO CONTINUARE ? (S/N)"
1190 GETZ$:IFZ$="N"GOTO1060
1200 IFZ$<>"S"GOTO1190
1210 PRINT"□                                      □":RETURN
```

```
100 REM ********************
110 REM      LISTATO N 5
120 REM ********************
130 REM   LEGGE SEQ. DISCO
140 REM ********************
150 POKE53280,0:POKE53281,0:PRINTCHR$(14)
160 OPEN15,8,15:INPUT"▮▮CRIVI IL NOME DEL FILE: ";F$
170 OPEN1,8,2,F$:INPUT#15,SX:IFSX<>0THENPRINT"ERRORE N.";SX:GOTO210
180 GET#1,A$:SS=ST:IFA$=CHR$(13)THENPRINT"←":GOTO200
190 PRINTA$;
200 IFSS=0THEN180
210 CLOSE1:CLOSE15
```

```
100 REM *******************
110 REM      LISTATO N 6
120 REM *******************
130 REM   LEGGE REL.DISCO
140 REM *******************
150 POKE53280,0:POKE53281,0:PRINTCHR$(14)
160 OPEN15,8,15:INPUT"▮/OME DEL FILE ";F$:F$=F$+".REL":LE=1
170 OPEN1,8,2,F$:INPUT#15,SS:IFSS<>0THENPRINT"ERRORE N.";SS:GOTO320
180 RC$=CHR$(80)+CHR$(98)+CHR$(1)+CHR$(0)+CHR$(LE):PRINT#15,RC$
190 INPUT#15,SS:IFSS=51THENLE=LE-2:GOTO210
200 LE=LE+1:GOTO180
210 INPUT"▮LECORD /O. (DA,A) ";I,L:I=I+1:L=L+1
220 RH=INT(I/256):RL=I-(RH*256)
230 RC$=CHR$(80)+CHR$(98)+CHR$(RL)+CHR$(RH)+CHR$(0):PRINT#15,RC$
240 INPUT#15,SS:IFSS=50 THEN PRINT"—INE DEL FILE":GOTO320
250 FORJ=1TOLE:GET#1,A$:IFA$=""THENA$=CHR$(0)
260 IFJ=1THENIFA$=CHR$(254)THENPRINT"╳╭┌│┌":GOTO290
270 X=ASC(A$):IFX<32THENX=X+64
280 PRINTCHR$(X);: NEXTJ
290 PRINT:FORK=1TO80:PRINT"-";:NEXT K
300 I=I+1:IFI>LTHEN320
310 GOTO 220
320 CLOSE1:CLOSE15:END
```

```
100 REM ******************
110 REM     LISTATO N 7
120 REM ******************
130 REM   LEGGE DATABASE
140 REM ******************
150 POKE53280,0:POKE53281,0:PRINTCHR$(14)
160 OPEN15,8,15:PRINT"▮/OME DEL FILE ,—AMPO # - ";:INPUTF$,F:P=1
170 X$=F$+".KY"+RIGHT$(STR$(F+100),2):OPEN1,8,2,X$
180 INPUT#15,SS,SS$,A,B:IFSS<>0THENPRINTSS;SS$;A;B:GOTO340
190 PRINT#15,CHR$(80)CHR$(98)CHR$(1)CHR$(0)CHR$(0)
```

```
200 GET#1,A$:IFA$=""THENA$=CHR$(0)
210 GET#1,B$:IFB$=""THENB$=CHR$(0)
220 GET#1,C$:IFC$=""THENC$=CHR$(0)
230 X=ASC(C$):N=ASC(A$)+(ASC(B$)*256)
240 D=INT(40/(X+5))
250 FORI=2TON+1:RH=INT(I/256):RL=I-(RH*256):C=I-2
260 PRINT#15,CHR$(80)CHR$(98)CHR$(RL)CHR$(RH)CHR$(0)
270 IFINT(C/D)=(C/D)THENPRINT:GOTO 290
280 PRINT"   ";
290 FORJ=1TOX-3:GET#1,A$:PRINTA$;:NEXT
300 GET#1,R$:IFR$=""THENR$=CHR$(0)
310 GET#1,S$:IFS$=""THENS$=CHR$(0)
320 R=ASC(R$)+(ASC(S$)*256):XR$=RIGHT$(STR$(R+10000),4)
330 PRINT"(";XR$;")";:NEXT
340 CLOSE1:CLOSE15:END
```

```
100 REM ********************
110 REM      LISTATO N 8
120 REM ********************
130 REM  LEGGE HEADER FILE
140 REM ********************
150 POKE53280,0:POKE53281,0:PRINTCHR$(14)
160 DIMJ(300)
170 OPEN15,8,15:INPUT"▙▜/OME DEL FILE ";N$:F=1:OPEN4,3:PRINT"▙"
180 PRINT#4,"¬NIZIO FILE '";N$;".HDR' CAMPO DATI":PRINT#4
190 OPEN1,8,2,N$+".HDR,S,R":INPUT#15,SS,SS$,A,B:IFSS>=20 THEN 290
200 GOSUB410:N1=X:GOSUB410:N2=(X*256)+N1:GOSUB410
210 PRINT#4,"LUNGH. DEL _ECORD =";N2;" PER";X;" CAMPI."
220 GOSUB410:PRINT#4,"/UMERO DI PAG. =";X:GOSUB410
230 PRINT#4,"LUNGHEZZA FISICA=";X;" RECORDS";:GOSUB410
240 PRINT#4," DI";X;" CARATTERI.":FORK=1TO5:GOSUB410:NEXT
250 PRINT#4,:PRINT#4,"¬ CAMPI SONO DEFINITI COSI':"
260 PRINT#4,"================================"
270 GOSUB410:PG=X:PRINT#4,"¬ER PAG. #";X:PRINT#4,"--------------"
280 FORU=1TO4:GOSUB410:NEXT:I=1
290 GOSUB410:J(I)=X:I=I+1:IFX<>0THEN290
300 GOSUB410:N1=X:GOSUB410:N3=X:N2=(X*256)+N1:GOSUB410:TP=X
310 GOSUB410:GOSUB410:PN=X:IFN3=128THEN280
320 IFPN=PGTHEN340
330 PG=PN:PRINT#4,:PRINT#4,"¬ER PAG. #";PN:PRINT#4,"--------------"
340 I=1:Q=0:PRINT#4,"¬AMPO";F;"(";:IFJ(I)=0THEN380
350 IFJ(I)<32THENPRINT#4,CHR$(J(1)+64);:Q=1
360 IFJ(I)>=32THENPRINT#4,CHR$(J(I));:Q=1
370 I=I+1:IFJ(I)<>0THEN350
380 IFQ=0THENPRINT#4,"-CONT'D-";
390 PRINT#4,")";" TIPO =";TP;" LUNGH. =";N2:F=F+1:GOTO280
400 END
410 GET#1,A$:SS=ST:IFA$=""THENA$=CHR$(0)
420 IFSS=64THENCLOSE1:CLOSE15:END
430 X=ASC(A$):RETURN
440 PRINTSS;SS$;A;B:CLOSE1:CLOSE15
```

```
100 REM ***********************************
103 REM             LISTATO N 9
105 REM ***********************************
110 REM         ARCHIVIO RANDOM
120 REM ***********************************
130 GH=53280:JK=13:JL=0:GJ=646
140 POKEGH,JK:POKEGH+1,JL:POKEGJ,JK
150 SP$="                                  "
160 CH$=CHR$(13):LIM$=CHR$(99)+CHR$(99)+CHR$(99)
170 DIMD$(14):DIMY$(14):DIML(14)
180 D$(1)="COGNOME: "
190 D$(2)="NOME: "
200 D$(3)="INDIRIZZO: "
210 D$(4)="LOCALITA': "
220 D$(5)="PROVINCIA: "
230 D$(6)="TELEFONO: "
240 D$(7)="LUOGO NASCITA: "
250 D$(8)="DATA NASCITA: "
260 D$(9)="TITOLO DI STUDIO: "
270 D$(10)="OCC. ATTUALE: "
280 D$(11)="OCC. PRECEDENTE: "
290 D$(12)="STATO CIVILE: "
300 D$(13)="NOTA 1: "
310 D$(14)="NOTA 2: "
320 L(1)=20:L(2)=15:L(3)=30:L(4)=25:L(5)=4
330 L(6)=10:L(7)=20:L(8)=8:L(9)=20:L(10)=20:L(11)=20:L(12)=8
340 L(13)=20:L(14)=20
350 OPEN15,8,15
360 PRINT"██          ARCHIVIO RANDOM DISCO          █":PRINT
370 PRINT"█       █ 1 █_________INIZIO EX NOVO"
380 PRINT"█       █ 2 █_________AGGIORNAMENTO"
390 PRINT"█       █ 3 █_________LISTA PRINCIPALE
400 PRINT"█       █ 4 █_________CREAZ.IND.SECOND.
410 PRINT"█       █ 5 █_________LISTA SECONDARIA
420 PRINT"█       █ 9 █_________ F I N E
430 INPUT"█  COSA SCEGLI";X
440 IFX<1ANDX>5ANDX<>9THEN430
450 PRINT:GOSUB960
460 PRINT"INSERISCI DISCO DATI"
470 GETR$:IFR$=""THEN470
480 IFX=1THEN530
490 PRINT#15,"I"
500 GOSUB860:N=K:PRINT"PRESENTI";K;"RECORD"
510 PRINT"DATA ULTIMO AGG.        ";G1$;"/";M1$;"/";"A1$
520 IFX<>2THEN540
530 PRINT"QUANTI RECORD";:INPUT N
540 DIMC$(N),T%(N),S%(N)
550 IFX<>1THENGOSUB900
560 IFX=9THEN2080
570 ONXGOTO1010,1150,1770,1920,2010
580 STOP
590 REM INGRESSO DATI
600 PRINT"█INGRESSO DATI"
610 PRINT"PER USCIRE $ PER COGN. SE MANC. DATI METTI 0"
```

```
620 PRINTD$(1);:INPUTY$(1)
630 IFY$(1)="$"THENW=1:RETURN
640 FORJ=2TO14:PRINTD$(J);:INPUTY$(J):NEXTJ
650 FORJ=1TO14:Y$(J)=LEFT$(Y$(J)+SP$,L(J)):NEXTJ:RETURN
660 REM INDICE PRINCIPALE
670 PRINT#15,"S0:INDI1":PRINT#15,"I":OPEN10,8,10,"INDI1,S,W":GOSUB930
680 FORJ=1TOK:PRINT#10,C$(J);CH$;T%(J);CH$;S%(J);CH$;:GOSUB930:NEXTJ
690 CLOSE10:RETURN
700 REM RECORD E ELEM. INDICE
710 FORJ=1TO14:PRINT#11,Y$(J);CH$;:GOSUB930:NEXTJ:RETURN
720 REM TRACCIA E SETTORE
730 PRINT#15,"B-A:"0;T;S
740 INPUT#15,EN,EM$,ET,ES
750 IFEN=0THENRETURN
760 IFEN<>65THEN950
770 T=ET:S=ES:GOTO730
780 REM PUNTATORE
790 PRINT#15,"B-P:"11;1:GOSUB930:RETURN
800 REM SCRITTURA
810 PRINT#15,"U2:"11;0;T;S:GOSUB930:RETURN
820 REM LETTURA
830 PRINT#15,"I":OPEN11,8,11,"#":GOSUB930
840 PRINT#15,"U1:"11;0;T;S:GOSUB930:GOSUB790
850 FORJ=1TO14:INPUT#11,Y$(J):GOSUB930:NEXTJ:CLOSE11:RETURN
860 REM LETT. DATI DISCO
870 OPEN11,8,11,"#":GOSUB930:PRINT#15,"U1:"11;0;1;0:GOSUB930
880 GOSUB780:INPUT#11,R$,K,T,S:GOSUB930
890 G1$=LEFT$(R$,2):M1$=MID$(R$,3,2):A1$=RIGHT$(R$,2):CLOSE11:TT=T:SS=S:RETURN
900 REM LETT. INDICE
910 OPEN10,8,10,"INDI1,S,R":GOSUB930
920 FORJ=1TOK:INPUT#10,C$(J),T%(J),S%(J):GOSUB930:NEXTJ:CLOSE10:RETURN
930 INPUT#15,EN,EM$,ET,ES
940 IFEN=0THENRETURN
950 PRINT"█ERRORE DISCO":PRINTEN,EM$,ET,ES:CLOSE15:RUN
960 REM DATA DISCO
970 PRINT"DATA PER DISCO"
980 INPUT"  GG,MM,AA██████████";G$,M$,A$:RETURN
990 GETR$:IFR$=""THEN990
1000 RETURN
1010 REM INIZIALIZZAZIONE DEL DISCO
1020 PRINT"NOME DEL DISCO";:INPUTN$:T=1:S=0
1030 PRINT#15,"N0:"+N$+",99"
1040 CLOSE15:OPEN15,8,15:PRINT#15,"I"
1050 REM DATA AGGIORNAMENTO E NUM. RECORD
1060 OPEN11,8,11,"#":GOSUB930:GOSUB730:GOSUB790
1070 K=0:PRINT#11,G$+M$+A$;CH$;K;CH$;2;CH$;0;CH$;:GOSUB930
1080 PRINT#15,"U2:"11;0;T;S:GOSUB930:K=1:W=0:T=2:S=0
1090 GOSUB590
1100 IFW=1THENK=K-1:CLOSE11:GOTO2080
1110 GOSUB720:GOSUB780:GOSUB700:C$(K)=Y$(1)+Y$(2)
1120 T%(K)=T:S%(K)=S:GOSUB800:K=K+1
1130 IFK>NTHENK=K-1:PRINT"FINITO SPAZIO ASSEGNATO":CLOSE11:GOTO2080
1140 GOTO1090
1150 REM AGGIORNAMENTO
```

```
1160 PRINT"          AGGIORNAMENTO ARCHIVIO           "
1170 PRINT" 1-CORREZIONE"
1180 PRINT" 2-AGGIUNTA ELEM."
1190 PRINT" 3-CANCELL. ELEM."
1200 PRINT" 9-     F I N E
1210 INPUT"COSA SCEGLI";X:IFX=9THEN2080
1220 IFX<1ORX>3THEN1160
1230 IFX=2THEN1570
1240 IFX=3THEN1610
1250 REM CORREZIONE
1260 PRINT"";D$(1);:INPUTY$(1)
1270 IFY$(1)="$"THEN1160
1280 PRINTD$(2);:INPUTY$(2)
1290 Y$(1)=LEFT$(Y$(1)+SP$,L(1))
1300 Y$(2)=LEFT$(Y$(2)+SP$,L(2))
1310 INPUT"QUALE OCCORRENZA";X
1320 IFX<=0THEN1310
1330 FORJ=1TOK:IFC$(J)=Y$(1)+Y$(2)THEN1370
1340 NEXTJ
1350 J=J-1:PRINT"NON TROVATO";Y$(1);"   ";Y$(2)
1360 GOSUB990:GOTO1260
1370 IFX<>1THENX=X-1:GOTO1340
1380 T=T%(J):S=S%(J)
1390 I=J:J=K:NEXTJ
1400 GOSUB820
1410 J1=1:J2=7:GOSUB1440
1420 J1=8:J2=14:GOSUB1440
1430 GOTO1540
1440 PRINT"";:FORJ=J1TOJ2:PRINTJ;"   ";D$(J);"   ";Y$(J):NEXTJ
1450 PRINT"QUALE CAMPO(0=USCITA)"
1460 INPUTX
1470 IFX=0THEN1510
1480 IFX>J2ORX<J1THENPRINT"":GOTO1460
1490 INPUTY$(X):Y$(X)=LEFT$(Y$(X)+SP$,L(X))
1500 PRINT"":GOTO1460
1510 PRINT"";:FORJ=J1TOJ2:PRINTJ;"   ";D$(J);"   ";Y$(J):NEXTJ
1520 INPUT"TUTTO BENE";X$:IFX$<>"S"THEN1450
1530 RETURN
1540 PRINT#15,"I":OPEN11,8,11,"#":GOSUB780:GOSUB700
1550 C$(I)=Y$(1)+Y$(2):T%(I)=T:S%(I)=S
1560 GOSUB800:CLOSE11:GOTO1250
1570 REM AGGIUNTA
1580 OPEN11,8,11,"#":GOSUB930:PRINT"AGGIUNTA":PRINT"PRESENTI";K;"RECORD"
1590 PRINT"PUOI AGGIUNGERE";N-K;"RECORD":GOSUB990
1600 T=TT:S=SS:W=0:K=K+1:GOTO1090
1610 REM CANCELLAZIONE
1620 PRINT"CANC. ELEM.":M=0
1630 PRINTD$(1);:INPUTY$(1)
1640 IFY$(1)="$"THEN1750
1650 PRINTD$(2);:INPUTY$(2)
1660 Y$(1)=LEFT$(Y$(1)+SP$,L(1))
1670 Y$(2)=LEFT$(Y$(2)+SP$,L(2))
1680 FORJ=1TOK:IFC$(J)=Y$(1)+Y$(2)THEN1720
1690 NEXTJ
```

```
1700 PRINT"NON TROVATO";Y$(1);" ";Y$(2)
1710 GOSUB990:GOTO1630
1720 X=L(1)+L(2):C$(J)=LEFT$(LIM$+SP$+SP$,X)
1730 M=M+1:T=T%(J):S=S%(J)
1740 PRINT#15,"B-F:"0;T;S:GOSUB930:GOTO1630
1750 REM SIST. INDICE
1760 GOSUB2190:K=K-M:GOTO2120
1770 REM LISTA
1780 T$="LISTA PER INDICE PRIM."
1790 PRINT"ACCENDI STAMPANTE       PREMI UN TASTO"
1800 GETR$:IFR$=""THEN1800
1810 PRINT"▙";T$:OPEN4,4
1820 PRINT#4:PRINT#4:PRINT#4,T$
1830 FORJ=1TO10:PRINT#4:NEXTJ
1840 L=2:FORI=1TOK:T=T%(I):S=S%(I):GOSUB820
1850 PRINT#4,Y$(1);"  ";Y$(2)
1860 PRINT#4,Y$(3);"  ";Y$(4);"  ";Y$(5)
1870 FORM=6TO14:PRINT#4,D$(M);Y$(M):NEXTM
1880 PRINT#4:PRINT#4
1890 IFL=5THENPRINT#4:L=0
1900 L=L+1:NEXTI
1910 CLOSE4:GOTO2160
1920 REM IND. SEC.
1930 PRINT"▙INDICE SECONDARIO"
1940 INPUT"■QUALE CAMPO";X$
1950 X=VAL(X$):IFX<2ORX>14THENGOTO1940
1960 GOSUB660
1970 FORI=1TOK:T=T%(I):S=S%(I):GOSUB820:C$(I)=Y$(X):NEXTI
1980 GOSUB1210:GOSUB900
1990 PRINT"FINITO IND. SEC."
2000 GOTO2160
2010 REM LISTA IND. SEC.
2020 PRINT"▙LISTA IND. SEC."
2030 OPEN10,8,10,"INDI2,S,R":GOSUB930
2040 FORJ=1TOK:INPUT#10,C$(J),T%(J),S%(J):GOSUB930:NEXTJ
2050 CLOSE10:GOSUB2320:GOSUB1210
2060 T$="LISTA IND. SEC."
2070 GOTO1790
2080 REM CHIUSURA
2090 PRINT"CHIUSURA ARCHIVIO"
2100 PRINT"IND. PRINC. DA ORDIN.":INPUTR$
2110 IFR$="S"THENGOSUB2190
2120 GOSUB660:OPEN11,8,11,"#":GOSUB930
2130 PRINT#15,"B-P:"11,1
2140 PRINT#11,G$+M$+A$;CH$;K;CH$;T;CH$;S;CH$;:GOSUB930
2150 PRINT#15,"U2:"11;0;1;0:GOSUB930:CLOSE11
2160 PRINT"FINITO AGGIORNAMENTO"
2170 PRINT"SONO PRESENTI";K;"RECORD"
2180 CLOSE15:STOP
2190 REM ORDINAMENTO CRESCENTE
2200 L=K-1
2210 W=0
2220 FORJ=1TOL
2230 IFC$(J)<=C$(J+1)THEN2280
```

```
2240 R$=C$(J):C$(J)=C$(J+1):C$(J+1)=R$
2250 X=T%(J):T%(J)=T%(J+1):T%(J+1)=X
2260 X=S%(J):S%(J)=S%(J+1):S%(J+1)=X
2270 W=1
2280 NEXTJ
2290 IFW=0THENRETURN
2300 IFL=1THENRETURN
2310 L=L-1:GOTO2210
2320 REM ORDINAMENTO CRESCENTE
2330 L=K-1
2340 W=0
2350 FORJ=1TOL
2360 IFC$(J)>=C$(J+1)THEN2410
2370 R$=C$(J):C$(J)=C$(J+1):C$(J+1)=R$
2380 X=T%(J):T%(J)=T%(J+1):T%(J+1)=X
2390 X=S%(J):S%(J)=S%(J+1):S%(J+1)=X
2400 W=1
2410 NEXTJ
2420 IFW=0THENRETURN
2430 IFL=1THENRETURN
2440 L=L-1:GOTO2340
```

```
100 REM **********************************
103 REM            LISTATO N 10
105 REM **********************************
110 REM            RECUPERATORE
120 REM         DI FILE CANCELLATI
140 REM **********************************
150 POKE53280,0:POKE53281,0
160 PRINT"███            RECUPERO FILE           ███"
170 DIM FE$(32),TF$(5):X$="":R$=""
180 FOR I=0 TO 5:READ TF$(I):NEXT:CC=0
190 K=0:CH=8:S=1:TR=18:T=TR:DR$="0"
200 IN$="█TIPO   NOME FILE      TR.SET.LUNG."
210 PRINT"█ 1 █ RECUPERO FILE CANCELLATO█"
220 PRINT"█ 2 █ RECUPERO DI TUTTI I FILE CANCELLATI█"
230 PRINT"█ 3 █ CONVALIDA█"
240 PRINT"█ 4 █ VISUALIZZAZIONE DIRECTORY█"
250 PRINT"█ 5 █ CANCELLAZIONE FILE█":PRINT"█ 6 █ FINE█"
260 PRINT"██SPAZIO█ PER TORNARE AL MENU.█"
270 PRINT "██1:REC1;2:RECT;3:CONV;4:DIR;5:CANC;6:END ███";
280 GET R$:IF R$="" THEN 280
290 IF R$<"1" OR R$>"6" THEN 280
300 R=VAL(R$):PRINT R$:RT=0
310 ON R GOTO 490,1110,1150,1280,1550,1720
320 END:REM **************************
330 REM * LEGGE E STAMPA DIRECTORY **
340 REM ******************************
350 TT=ASC(FE$(1)):SS=ASC(FE$(2))
360 FT=ASC(FE$(0)) AND 127
```

```
370 IF FT>5 THEN PRINT "TIPO ERRATO":FT=5
380 TF$=TF$(FT)
390 LF=ASC(FE$(28))+256*ASC(FE$(29))
400 TT$=RIGHT$("  "+STR$(TT),3)
410 SS$=RIGHT$("  "+STR$(SS),3)
420 LF$=RIGHT$("     "+STR$(LF),5)
430 PRINT "■"TF$"■ "NF$;TAB(20);
440 IF SP THEN PRINT#3,"■"TF$"■ "NF$;
450 PRINT TT$;SS$;LF$
460 IF SP THEN PRINT#3,TT$;SS$;LF$
470 RETURN
480 REM ******************************
490 PRINT "■RICERCA DI UN CERTO FILE"
500 REM ******************************
510 GOSUB850
520 PRINT "NOME DEL FILE DA RECUPERARE"
530 INPUT NO$:L=LEN(NO$):FOR K=1 TO L
540 IF MID$(NO$,K,1)="*" THEN L=K-1:REMK=80
550 NEXTK:REMIF K<80 THEN L=16
560 NO$=LEFT$(NO$+"                ",L)
570 PRINT "STO CERCANDO    ■"NO$
580 REM *******************************
590 REM *RICERCA FILE NELLA DIRECTORY*
600 REM *******************************
610 T=TR:OPEN1,8,15:PRINT#1,"I"+DR$
620 S=1:OPENCH,8,CH,"#"+DR$
630 PRINT#1,"U1:"CH;DR;T;S
640 PRINT#1,"B-P:"CH;0:GET#CH,T$,S$
650 FOR FE=1 TO 8
660 GET R$:IF R$=" " THEN 780
670 FOR CC=0 TO 31:GET#CH,X$
680 IF X$="" THEN X$=CHR$(0)
690 FE$(CC)=X$:NEXT:NF$=""
700 FOR K=3 TO 18:NF$=NF$+FE$(K):NEXT
710 IF NO$=LEFT$(NF$,L) THEN 1180
720 NEXTFE
730 IF T$="" THEN T$=CHR$(0)
740 IF S$="" THEN S$=CHR$(0)
750 T=ASC(T$):S=ASC(S$)
760 IF T<>0 THEN 630
770 PRINT "■NON TROVATO■"NF$
780 CLOSECH:CLOSE1:CLOSE3:GOTO270
790 REM *******************************
800 REM ** GESTIONE ERRORI DA DISCO **
810 REM *******************************
820 INPUT#1,CE$,ER$,ET$,ES$:IF CE$="00" THEN RETURN
830 PRINT CE$,ER$,ET$;ES$:GOTO780
840 REM *******************************
850 REM ******   QUALE DRIVE   ********
860 REM *******************************
870 PRINT"DRIVE    ";DR$"■■■■";:INPUT DR$
880 IF DR$<>"0" AND DR$<>"1" THEN870
890 DR=VAL(DR$)
900 RETURN
```

```
910 REM ********************************
920 REM ***  SCELTA TIPO DI FILE   ***
930 REM ********************************
940 PRINT"0=DEL;1=SEQ;2=PRG;3=USR;4=REL;5=>MENU"
950 INPUT"TIPO   2";R:IF R<0 OR R>5 THEN950
960 IF R=5 THEN 780
970 PO=2+32*(FE-1):TP=R+128
980 PRINT#1,"U1:"CH;DR;T;S
990 PRINT#1,"B-P:"CH;PO
1000 PRINT#CH,CHR$(TP);
1010 PRINT#1,"U2:"CH;DR;T;S
1020 RETURN
1030 REM ********************************
1040 REM ** IL FILE VA RECUPERATO ?? **
1050 REM ********************************
1060 IF MID$(NF$,1,1)=CHR$(0) THEN1240
1070 INPUT"DA RECUPERARE (S/N/END)  N";R$
1080 IF R$="S" THEN GOSUB920:PRINT#1,"B-P:"CH;PO+32
1090 IF LEFT$(R$,1)="E" THEN 1240
1100 RETURN
1110 REM ********************************
1120 PRINT "RECUPERO TUTTI I FILE"
1130 RT=-1:GOTO1300
1140 REM ********************************
1150 PRINT "CONVALIDA ";:GOSUB850
1160 CLOSE1:OPEN1,8,15:PRINT#1,"V"+DR$
1170 GOSUB800:GOTO780
1180 REM ******************************
1190 PRINT"TROVATO IL FILE "NF$
1200 PRINT IN$
1210 GOSUB350
1220 PRINT "CHE TIPO ERA  ?"
1230 GOSUB920
1240 INPUT "CONVALIDA (S/N)   S";R$
1250 IF R$="S" THEN1150
1260 GOTO780
1270 REM ******************************
1280 PRINT"VISUALIZZAZIONE  DIRECTORY"
1290 REM ******************************
1300 GOSUB850
1310 SP=0
1320 GOSUB1610:PRINT IN$:S=1
1330 IF SP THEN PRINT#3,IN$
1340 REM ************
1350 OPEN1,8,15:PRINT#1,"I"+DR$
1360 OPENCH,8,CH,"#"+DR$:GOSUB800
1370 PRINT#1,"U1:"CH;DR;T;S
1380 PRINT#1,"B-P:"CH;0:GET#CH,T$,S$
1390 FOR FE=1 TO 8
1400 GET R$:IF R$=" " THEN1520
1410 FOR CC=0 TO 31:GET#CH,X$
1420 IF X$="" THEN X$=CHR$(0)
1430 FE$(CC)=X$:NEXT:NF$=""
1440 FOR K=3 TO 18:NF$=NF$+FE$(K):NEXT
```

```
1450 GOSUB350:IF RT AND (FT=0) THEN GOSUB1040
1460 IF FE$(3)=CHR$(0) THEN 1520
1470 NEXTFE
1480 IF T$="" THEN T$=CHR$(0)
1490 IF S$="" THEN S$=CHR$(0)
1500 T=ASC(T$):S=ASC(S$)
1510 IF T<>0 THEN 1370
1520 IF RT THEN 1240
1530 T=TR:GOTO780
1540 REM ****************************
1550 PRINT "█CANCELLAZIONE FILE":GOSUB850
1560 PRINT"NOME DEL FILE DA CANCELLARE"
1570 INPUT R$:CLOSE1:OPEN1,8,15
1580 PRINT#1,"S"+DR+":"+R$
1590 GOSUB800:GOTO780
1600 REM *****************************
1610 REM LEGGE NOME,IDENTIF. DISCHETTO
1620 REM *****************************
1630 PO=6:IF TR=18 THEN PO=144
1640 S=0:OPEN1,8,15,"I"+DR$:GOSUB800
1650 OPENCH,8,CH,"#":PRINT#1,"U1:"CH;DR;TR;S
1660 PRINT#1,"B-P:";CH;PO:NF$=""
1670 FOR K=1 TO 20:GET#CH,X$:NF$=NF$+X$:NEXT
1680 PRINT DR$"  "NF$:IF SP THEN PRINT#3,DR$"  "NF$
1690 CLOSECH:CLOSE1:RETURN
1700 DATA DEL,SEQ,PRG,USR,REL,XXX
1710 REM *****************************
1720 PRINT TAB(18)"█FINE":CLOSE1:CLOSECH:CLOSE3:END
```

```
100 REM *******************
110 REM      LISTATO N 11
120 REM *******************
130 REM  TROVA ERRORI 1541
140 REM *******************
150 CLR:PRINTCHR$(14)
160 DIMA(4):DIMT(4):DIMFT(4)
170 T(1)=1:FT(1)=17:T(2)=18:FT(2)=24:T(3)=25:FT(3)=30:T(4)=31:FT(4)=35
180 A(1)=20:A(2)=18:A(3)=17:A(4)=16
190 POKE53280,0:POKE53281,0:PRINT"██"
200 PRINT"┌──────────────────────┐"
210 PRINT"| █TROVA ERRORI 1541 █|        |"
220 PRINT"├──────────────────────┘       |"
230 PRINT"| █(C) EDISOFT SRL MILANO    █|"
240 PRINT"└──────────────────────────────┘"
250 PRINT" ███CHERMO O ██RINTER  ";
260 GET V$ :IFV$<>"S"ANDV$<>"P"THENGOTO260
270 IFV$="P"THEN OPEN4,4
280 PRINT"██":OPEN15,8,15:OPEN5,8,5,"#"
290 GOSUB 420
```

```
300 FOR IM=1TO4
310 FOR TR=T(IM)TOFT(IM)
320 FOR SE=0TOA(IM)
330 TR$=STR$(TR):SE$=STR$(SE)
340 TES$="B-R,5,0,"+TR$+","+SE$
350 PRINT#15,TES$
360 GET#5,Q$:GOSUB 420
370 PRINT"□TRACCIA :";TR;" SETTORE :     ■■■";SE
380 NEXT SE,TR,IM
390 IFV$="P"THENCLOSE4
400 CLOSE15,5
410 END
420 INPUT#15,E$,S$,L$,G$
430 IFE$<>"00"THEN450
440 RETURN
450 PRINT"ERRORE:"
460 PRINTE$;" ";S$;" ";L$;" ";G$;"■"
470 IFV$="P"THENPRINT#4,E$;" ";S$;" ";L$;" ";G$;"■"
480 RETURN
```

```
500 REM *******************
504 REM    LISTATO N 12
505 REM *******************
510 REM CREATORE DI ERRORI
520 REM   PER C=64 E 1541
530 REM *******************
560 CLR
570 BY(0)=0
580 CA(0)=1
590 CN(0)=1
600 REM
610 REM *******************
620 REM PARAMETRI
630 BY(2)=0
640 CA(2)=15
650 CN(2)=40
660 BY(3)=0
670 CA(3)=128
680 CN(3)=25
690 K$(0)="■"
700 FORX=0TO88:READD:M1$=M1$+CHR$(D):NEXT
710 FORX=0TO67:READD:M2$=M2$+CHR$(D):NEXT
720 OPEN15,8,15
730 OPEN2,8,2,"#0"
740 OPEN3,8,3,"#1"
750 PRINT"□":POKE53280,0:POKE53281,0
760 PRINT"■■■          CREATORE DI ERRORI          ■"
770 PRINT"□-------------------------------------"
780 S$=" READ ERROR"
790 PRINT"■■■■■■ F 1 ■_____SELEZIONA L'ERRORE"
800 PRINT"■■■■■ F 3 ■_____TRACCIA E SETTORE"
810 PRINT"■■■■■ F 5 ■_____FINE DEL PROGRAMMA■■■"
```

```
820 PRINT"■"K$(0)"20"S$
830 PRINT"■"K$(1)"21"S$
840 PRINT"■"K$(2)"22"S$
850 PRINT"■"K$(3)"23"S$
860 GETT$:IFT$=""THEN860
870 IFT$="■"THENK$(S)="":S=3ANDS+1:K$(S)="■"
880 IFT$="■"THEN910
890 IFT$="■"THENCLOSE15:END
900 GOTO760
910 INPUT"■■TRACCIA :";TR
920 IFS=1THENSE=0:PRINT"■DA QUALE SETTORE PARTO":GOTO940
930 INPUT"■SETTORE";SE
940 PRINT#15,"U1:"2;0;TR;SE
950 INPUT#15,A,B$,C,D
960 PRINT"■";A;B$;C;D
970 IFA<>0THENPRINT"■ !DISK ERROR!":GOTO1110
980 PRINT#15,"B-P:"3;0
990 IFS=1THENPRINT#3,M2$:GOTO1010
1000 PRINT#3,M1$
1010 TX=20
1020 IFTR>17THENTX=18
1030 IFTR>24THENTX=17
1040 IFTR>30THENTX=16
1050 SX=SE-1:IFSX<0THENSX=TX
1060 MC$="M-E"+CHR$(3)+CHR$(4)+CHR$(TR)+CHR$(SX)+CHR$(CA(S))+CHR$(CN(S))+CHR$(BY
(0))
1070 PRINT#15,MC$
1080 PRINT#15,"U1:"2;0;TR;SE
1090 INPUT#15,A,B$,C,D
1100 PRINT"■";A;B$;C;D
1110 GETT$:IFT$=""THEN1110
1120 GOTO750
1130 DATA76,22,4,173,5,2,174,6,2,133,8,134,9,169,224,133,1,165,1,48,252,96
1140 DATA32,10,245,44,0,28,16,251,32,86,245,174,8,2,172,7,2,184,80,254,136
1150 DATA208,250,169,255,141,3,28,173,12,28,41,31,9,192,141,12,28,173,9,2,141,1
1160 DATA28,184,80,254,202,208,250,173,12,28,9,224,141,12,28,169,0,141,3,28,76,1
05,249,0
1170 DATA76,22,4,173,5,2,174,6,2,133,8,134,9,169,224,133,1,165,1,48,252
1180 DATA96,169,255,141,3,28,173,12,28,41,31,9,192,141,12,28,169,0,141,1,28,168,
170,80
1190 DATA254,232,208,251,200,208,247,173,12,28,9,224,141,12,28,169,0,141,3,28,76
,105,249
```

```
6000 REM *********************
6004 REM      LISTATO N 13
6005 REM *********************
6010 REM      LEGGE-SCRIVE
6020 REM SUL BLOCCO DESIDERATO
6040 REM *********************
6070 OPEN15,8,15:OPEN8,8,8,"#":POKE53280,0:POKE53281,0
6080 INPUT"███TRACCIA, SETTORE: ";T,S
6090 PRINT"VUOI █S█CRIVERE O █L█EGGERE?"
6100 PRINT"██(*) PER FINIRE"
6110 GETF$
6120 IFF$="S"THEN6160
6130 IFF$="L"THEN6230
6140 IFF$="*"THEN6340
6150 GOTO6110
6160 PRINT"█STO SCRIVENDO NEL BLOCCO ";S" DELLA TRACCIA ";T;"█"
6170 PRINT#15,"B-P:";8;0
6180 FORI=0TO255:PRINT#8,"◢";:PRINT"█"I"██◢";:NEXT:PRINT
6190 PRINT#15,"U2:";8;0;T;S
6200 GOSUB6300
6210 PRINT#15,"B-P:";8;0
6220 FORI=0TO255:PRINT#8,"-";:NEXT
6230 PRINT"█LEGGO IL BLOCCO ";S;" DI TRACCIA ";T;"█"
6240 PRINT#15,"U1:";8;0;T;S
6250 GOSUB6300
6260 PRINT#15,"B-P:";8;0
6270 FORI=0TO255
6280 GET#8,A$:IFA$<>""THENPRINT"█"I"██"A$;
6290 NEXT:PRINT:GOTO6080
6300 INPUT#15,EN,EM$,ET,ES
6310 IFEN=0THENRETURN
6320 PRINT"█ERROR STATUS:"EN;EM$;ET;ES"█"
6330 RETURN
6340 OPEN1,8,15,"U+
6350 SYS64738
```

```
1000 REM******************
1010 REM     LISTATO N 14
1011 REM ******************
1012 REM     TEST ERRORE
1013 REM ******************
1020 D$="0"
1030 OPEN15,8,15,"I"+D$
1040 OPEN 2,8,2,"#"
1050 T=35:S=12:REM
1060 PRINT#15,"U1:2,"D$;T;S:GOSUB1080
1070 CLOSE2:CLOSE15:END
1080 INPUT#15,EN,EM$,ET,ES:IF EN=0 THEN PRINT"OK!":RETURN
1090 PRINT"█DISK ERROR█"EN,EM$,ET,ES
1100 IFEN=23THENPOKE53280,0:REM
1110 RETURN
1120 END
```

```
2000 REM ******************
2010 REM   LISTATO N 14 A
2020 REM ******************
2030 REM   TEST DIRECTORY
2040 REM ******************
2070 OPEN2,8,15
2080 PRINT"▢"
2090 OPEN1,8,0,"$0"
2100 GET#1,A$,B$
2110 GET#1,A$,B$
2120 GET#1,A$,B$
2130 C=0
2140 IF A$<>"" THEN C=ASC(A$)
2150 IF B$<>"" THEN C=C+ASC(B$)*256
2160 PRINTMID$(STR$(C),2);TAB(3);"■";
2170 GET#1,B$:IF ST<>0 THEN 1000
2180 IF B$<>CHR$(34) THEN 2170
2190 GET#1,B$:IF B$<>CHR$(34)THEN PRINTB$;:N$=N$+B$:GOTO2190
2200 GET#1,B$:IF B$=CHR$(32) THEN 2200
2210 C$=C$+B$:GET#1,B$:IF B$<>""  THEN 2210
2220 PRINTC$:CLOSE1:CLOSE2:REM** IF C$<>NOME ORIGINALE DISTRUGGI IL PROGRAMMA!!!
```

```
3000 REM ******************
3010 REM    LISTATO N 15
3020 REM ******************
3030 REM     TEST NOTCH
3040 REM ******************
3070 D$="0"
3080 CLOSE15:OPEN15,8,15
3090 CLOSE2:OPEN2,8,2,"#"
3100 PRINT#15,"U2:2,"D$;18;18
3110 INPUT#15,EN,EM$,ET,ES
3120 IF EN=0 THEN 3220
3130 IFEN<>26THEN3220
3140 PRINT#15,"U+
3150 POKE53280,0:POKE53281,0:PRINTCHR$(14)
3160 PRINT"▢■■■■■  ●UESTO PROGRAMMA NON FUNZIONA SE IL"
3170 PRINT"  DISCO E' STATO PROTETTO CON UNA LIN-"
3180 PRINT"  GUETTA ADESIVA!!! ■■    |OGLILA!!!    ■"
3190 PRINT"■■■■■■■■■■■■■■■■■■    ¬REMI UN TASTO PER CONTINUARE      ■"
3200 GETA$:IFA$=""THEN3200
3210 GOTO3070
3220 CLOSE2:CLOSE15
3230 PRINTEN,EM$,ET,ES
3240 END
```

```
4000 REM *****************
4010 REM   LISTATI N 16
4020 REM *****************
4030 REM    DESTROY 1.0
4040 REM *****************
4050 POKE808,225:POKE775,200
4060 PRINTCHR$(14)
4070 OPEN1,8,15,"N0:NO COPY.......!!
4080 CLOSE1
4090 PRINT"                 "
4100 POKE53280,0:POKE53281,0
4110 PRINT"   ♠TTENDI QUALCHE SECONDO MENTRE CARICO  "
4120 SAVE"+++++++++++++++",8
4130 SAVE"----------------",8
4140 SAVE"QUESTO PROGRAMMA",8
4150 SAVE"           !  ",8
4160 SAVE" -------------- ",8
4170 SAVE"                ",8
4180 SAVE" (QUI VENGONO   ",8
4190 SAVE" SCRITTE PAROLE ",8
4200 SAVE" IMPRONUNCIABILI",8
4210 SAVE" A VOSTRA       ",8
4220 SAVE"    DISCREZIONE)",8
4230 SAVE"AH AH AH AH !!!!",8
4240 OPEN1,8,15,"V0"
4250 CLOSE1
4260 PRINTCHR$(14):POKE53280,0:POKE53281,0
4270 PRINT"     -I HAI PROVATO, E STAVOLTA TI E' ANDATA"
4280 PRINT"  MOLTO MALE.            "
4290 PRINT"                              ♠ − − ╮ ┌ !!!     "
4300 FORI=0TO3000:NEXTI
4310 PRINT" ":PRINT"                    STAI PIU' ATTENTO!  "
4320 SYS64739
```

```
10 REM ********************************
20 REM           LISTATO N 17
30 REM ********************************
40 REM        COPIA NASTRO-DISCO
50 REM ********************************
60 PRINT" ":POKE56,40:CLR
70 POKE53280,7:POKE53281,7:POKE646,6
80 Y1=5280:Y2=Y1+20
90 DI=8:SP=1309:SB=10240
100 BL$="                                                          "
110 BL$=BL$+"         "
120 PRINT"          COPIA NASTRO-DISCO           "
130 PRINT"  ┌─ COPIA FILE DA NASTRO ──────────────┐"
140 PRINT"│                                      │"
150 PRINT"└──────────────────────────────────────┘"
160 PRINT"┌─ TRACCIA SETTORE ────────────────────┐"
```

```
170 PRINT"│                                        │"
180 PRINT"└────────────────────────────────────────┘"
190 PRINT"┌─■■DISCO■■─────────────────────────────────┐"
200 PRINT"│                                        │"
210 PRINT"└────────────────────────────────────────┘"
220 PRINT"┌─■■STATUS DEL DISCO■■──────────────────────┐"
230 PRINT"│                                        │"
240 PRINT"└────────────────────────────────────────┘"
250 PRINT"┌─■■ESECUZIONE■■────────────────────────────┐"
260 PRINT"│                                        │"
270 PRINT"└────────────────────────────────────────┘"
280 PRINT"┌─■■INTERVENTO DELL'OPERATORE■■─────────────┐"
290 PRINT"│                                        │"
300 PRINT"└────────────────────────────────────────┘"
310 PRINT"■■■(C) EDISOFT SRL     - MILANO -         ■■■"
320 OP$="":GOSUB640:OP$="INSERISCI IL NOME DEL DISCO "
330 GOSUB670:OP$="DESTINAZIONE:":GOSUB660:LN=16:DN=11:RT=17:GOSUB530
340 LN=2:RT=35:IFB$=""THENB$="COPIA N-D":OP$=OP$+"     "+B$:GOSUB660
350 DN$=B$:PRINT"□"SPC(34)",":OP$="NUMERO DELL'ID DEL DISCO":GOSUB670
360 GOSUB530:DI$="":IFB$<>""THENDI$=","+B$
370 PRINT"□"SPC(35)B$:POKE198,0
380 BC$="INSERISCI IL DISCO-DESTINAZIONE":GOSUB870
390 GOSUB700:OP$="FORMATTO IL DISCO-DESTINAZIONE":GOSUB640
400 OPEN15,DI,15:INPUT#15,EN$,ED$,ET,ES$:IFEN$="00"THEN420
410 CLOSE15:OP$="":GOSUB640:GOSUB680:OP$="":GOTO320
420 N$="N0:"+DN$+DI$:PRINT#15,N$:INPUT#15,EN$,ED$,ET$,ES$
430 GOSUB680:IFEN$<>"00"THEN:OP$="":GOSUB640:GOTO320
440 OP$="PREMI PLAY SUL REGISTRATORE":GOSUB670
450 SYS(Y2):OP$="":GOSUB670
460 SB=256*PEEK(829)+PEEK(828)
470 EB=256*PEEK(831)+PEEK(830)
480 SN$="":FORA=833TO848:SN$=SN$+CHR$(PEEK(A)):NEXT
490 OP$=SN$:GOSUB650
500 GOSUB740:IF(TR+SC)THEN450
510 OP$="COPIA ULTIMATA":GOSUB640:OP$="":GOSUB700:GOSUB650:GOSUB660
520 BC$="TOGLI IL DISCO DAL DRIVE":POKE56,160:GOSUB870:GOTO60
530 B$="":PRINTLEFT$("■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■",DN)SPC(RT)"  ";
540 PRINT"■■";:FORA=1TO55:GETA$:IFA$<>""THEN560
550 NEXT:PRINT"■ ";:FORA=1TO55:IFA$=""THENNEXT:GOTO540
560 IFASC(A$)=13THENPRINT"■ ":RETURN
570 IFASC(A$)=133THEN60
580 IFASC(A$)<>20THEN610
590 IFLEN(B$)=0THENNEXT:GOTO540
600 B$=LEFT$(B$,LEN(B$)-1):PRINT"■ ■■ ";:GOTO540
610 IFA$<" "ORA$>"Z"THENNEXT:GOTO540
620 IFLEN(B$)<LNTHENB$=B$+A$:PRINT"■"A$"■";:GOTO540
630 NEXT:GOTO540
640 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■"BL$OP$:RETURN
650 PRINT"■■■■■■■■"BL$OP$:RETURN
660 PRINT"■■■■■■■■■■■■■■■■"BL$OP$:RETURN
670 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■"BL$OP$:RETURN
680 CLOSE2:CLOSE15:OP$=EN$+", "+ED$+","+ET$+","+ES$
690 PRINT"■■■■■■■■■■■■■■■■■■■■"BL$OP$:RETURN
700 PRINT"■■■■■■■■■■■■■■■■■■■■"BL$:RETURN
```

```
710 Z=INT(TR/10):POKESP,Z+48:Z=TR-10*Z:POKESP+1,Z+48
720 Z=INT(SC/10):POKESP+7,Z+48:Z=SC-10*Z:POKESP+8,Z+48
730 RETURN
740 OP$="LEGGO I DATI DAL BUFFER":GOSUB640
750 BP=SB:OPEN15,DI,15:OPEN2,DI,2,"#"
760 IFBP>=EBTHENCLOSE15:CLOSE2:RETURN
770 TR=PEEK(BP):SC=PEEK(BP+1):IFTR=18ANDSC=0THEN830
780 IF(TR=0ANDSC=0)THENCLOSE2:CLOSE15:RETURN
790 PRINT#15,"B-P:"2;0:BP=BP+2:HI=INT(BP/256):LO=BP-(HI*256):POKEY1+8,LO
800 POKEY1+9,HI:GOSUB710:SYS(Y1):INPUT#15,EN$,ED$,ET$,ES$
810 IFEN$<>"00"THEN910
820 PRINT#15,"U2";2;0;TR;SC:BP=BP+256:GOTO760
830 DN$=DN$+"                "
840 FORA=0TO15:POKE(BP+146+A),ASC(MID$(DN$,A+1,1)):NEXT
850 IFDI$=""THEN790
860 FORA=0TO1:POKE(BP+164+A),ASC(MID$(DI$,2+A,1)):NEXT:GOTO790
870 OP$=BC$:GOSUB670:FORA=1TO75:GETA$
880 IFA$=""THENNEXT:OP$="":GOSUB670:FORA=1TO75:GETA$:IFA$=""THENNEXT:GOTO870
890 OP$="":GOSUB670:IFA$<>"■"THENRETURN
900 GOTO60
910 GOSUB680:OP$="COPIA INCOMPLETA - RICOMINCIA DACCAPO":GOSUB640
920 GOSUB640:BC$="ERRORE DEL DISCO ILLEGGIBILE":GOSUB870:GOTO60
```

```
10 REM ********************************
20 REM              LISTATO N 18
30 REM ********************************
40 REM          COPIA DISCO-NASTRO
50 REM ********************************
60 PRINT"▢":POKE56,40:CLR
70 POKE53280,6:POKE53281,6:POKE646,1
80 Y1=6400:Y2=Y1+20:Y3=Y1+36
90 DI=8:SP=1309:SB=10240:BP=SB:EB=SB+64*258:FC=0
100 BL$="                                          ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"
110 BL$=BL$+"■■■■■■■■":DIMBM$(140),C(35):FORA=1TO35:READC(A):NEXT:GOSUB1070
120 DATA21,21,21,21,21,21,21,21,21,21,21,21
130 DATA21,21,21,21,21,19,19,19,19,19,19,19,18,18,18,18,18,18,17,17,17,17,17
140 PRINT"▢■■        COPIA DA DISCO A NASTRO          ■"
150 PRINT"■■■┌─■■COMANDO DI COPIA■■──────────────────┐"
160 PRINT"│                                          │"
170 PRINT"└──────────────────────────────────────────┘"
180 PRINT"┌─■■TRACCIA SETTORE■■──────────────────────┐"
190 PRINT"│                                          │"
200 PRINT"└──────────────────────────────────────────┘"
210 PRINT"┌─■■DISCO■■────────────────────────────────┐"
220 PRINT"│                                          │"
230 PRINT"└──────────────────────────────────────────┘"
240 PRINT"┌─■■STATUS DEL DISCO■■─────────────────────┐"
250 PRINT"│                                          │"
260 PRINT"└──────────────────────────────────────────┘"
270 PRINT"┌─■■ESECUZIONE■■───────────────────────────┐"
```

```
280 PRINT"|                                                         |"
290 PRINT"└─────────────────────────────────────────────────────────┘"
300 PRINT"┌──INTERVENTO DELL'OPERATORE──────────────────┐"
310 PRINT"|                                                         |"
320 PRINT"└─────────────────────────────────────────────────────────┘"
330 PRINT"(C) EDISOFT SRL      - MILANO -              "
340 OP$="CODICE DI OPERAZIONE DEL PROGR. ":GOSUB1030:LN=1:DN=5:RT=1
350 GOSUB890:IFB$=""THENB$="B"
360 PO=ASC(B$):IFB$="D"THENOP$="COPIA DIRETTA":GOTO390
370 IFB$<>"B"THEN340
380 OP$="BAM COPIA SELEZIONATA"
390 GOSUB1010:OP$="":GOSUB1000
400 POKE198,0
410 BC$="METTI IL DISCO SORGENTE NEL DRIVE":GOSUB1250
420 OP$="LETTURA BAM DAL DISCO SORGENTE":GOSUB1000
430 OPEN15,DI,15:OPEN2,DI,2,"#":PRINT#15,"U1";2;0;18;0:PRINT#15,"B-P:"2
440 FORA=1TO140:GET#2,A$:IFA$=""THENA$=CHR$(0)
450 BM$(A)=A$:NEXT:INPUT#15,EN$,ED$,ET$,ES$:IFEN$<>"00"THEN1290
460 PRINT#15,"B-P";2;144:SN$="":FORA=1TO20:GET#2,A$:SN$=SN$+A$:NEXT
470 FORA=1TO12:POKEY2+A+1,ASC(MID$(SN$,A,1)):NEXT
480 POKEY2+14,ASC(MID$(SN$,19,1))
490 POKEY2+15,ASC(RIGHT$(SN$,1))
500 CLOSE2:CLOSE15:SN$=LEFT$(SN$,16)+","+RIGHT$(SN$,2)
510 OP$="SORG. :              "+SN$
520 GOSUB1020:OP$="":GOSUB1000
530 BC$="VERIFICA IL DISCO SORGENTE   ":GOSUB1250
540 IFA$<>"N"THEN560
550 OP$="":GOSUB1000:GOSUB1020:GOSUB1060:GOTO410
560 OP$="LEGGE I DATA NEL BUFFER":GOSUB1000
570 IFPO=68THENFORA=1TO140:BM$(A)=CHR$(0):NEXT
580 OPEN15,DI,15:OPEN2,DI,2,"#"
590 FORTC=1TO140STEP4:IFC(((TC-1)/4+1)+0)=ASC(BM$(TC))THEN840
600 TR=(TC-1)/4+1:BY=255-ASC(BM$(TC+1))
610 IFBYAND128THENSC=7:GOSUB1110
620 IFBYAND64THENSC=6:GOSUB1110
630 IFBYAND32THENSC=5:GOSUB1110
640 IFBYAND16THENSC=4:GOSUB1110
650 IFBYAND8THENSC=3:GOSUB1110
660 IFBYAND4THENSC=2:GOSUB1110
670 IFBYAND2THENSC=1:GOSUB1110
680 IFBYAND1THENSC=0:GOSUB1110
690 BY=255-ASC(BM$(TC+2))
700 IFBYAND128THENSC=15:GOSUB1110
710 IFBYAND64THENSC=14:GOSUB1110
720 IFBYAND32THENSC=13:GOSUB1110
730 IFBYAND16THENSC=12:GOSUB1110
740 IFBYAND8THENSC=11:GOSUB1110
750 IFBYAND4THENSC=10:GOSUB1110
760 IFBYAND2THENSC=9:GOSUB1110
770 IFBYAND1THENSC=8:GOSUB1110
780 BY=255-ASC(BM$(TC+3))
790 IF(BYAND16)ANDC(TR)>20THENSC=20:GOSUB1110
800 IF(BYAND8)ANDC(TR)>19THENSC=19:GOSUB1110
810 IF(BYAND4)ANDC(TR)>18THENSC=18:GOSUB1110
```

```
820 IF(BYAND2)ANDC(TR)>17THENSC=17:GOSUB1110
830 IF(BYAND1)ANDC(TR)>16THENSC=16:GOSUB1110
840 NEXT:CLOSE2:CLOSE15:OP$="":GOSUB1000
850 POKEBP,0:POKEBP+1,0:BP=BP+2:HI=INT(BP/256):LO=BP-(HI*256)
860 POKE174,LO:POKE175,HI:GOSUB1200
870 OP$="COPIA ULTIMATA":GOSUB1000:OP$="":GOSUB1060:GOSUB1010:GOSUB1020
880 BC$="TOGLI IL DISCO DAL DRIVE":POKE56,160:GOSUB1250:GOTO60
890 B$="":PRINTLEFT$("█████████████████████████",DN)SPC(RT)" ";
900 PRINT"██";:FORA=1TO55:GETA$:IFA$<>""THEN920
910 NEXT:PRINT"█ ";:FORA=1TO55:IFA$=""THENNEXT:GOTO900
920 IFASC(A$)=13THENPRINT"█ ":RETURN
930 IFASC(A$)=133THEN60
940 IFASC(A$)<>20THEN970
950 IFLEN(B$)=0THENNEXT:GOTO900
960 B$=LEFT$(B$,LEN(B$)-1):PRINT"█ ██ ";:GOTO900
970 IFA$<" "ORA$>"Z"THENNEXT:GOTO900
980 IFLEN(B$)<LNTHENB$=B$+A$:PRINT"█"A$"█";:GOTO900
990 NEXT:GOTO900
1000 PRINT"████████████████████"BL$OP$:RETURN
1010 PRINT"██████"BL$OP$:RETURN
1020 PRINT"████████████"BL$OP$:RETURN
1030 PRINT"████████████████████████"BL$OP$:RETURN
1040 CLOSE2:CLOSE15:OP$=EN$+", "+ED$+","+ET$+","+ES$
1050 PRINT"█████████████████"BL$OP$:RETURN
1060 PRINT"█████████████████"BL$:RETURN
1070 FORA=SBTOEBSTEP258:POKEA,0:POKEA+1,0:NEXT:BP=SB:RETURN
1080 Z=INT(TR/10):POKESP,Z+48:Z=TR-10*Z:POKESP+1,Z+48
1090 Z=INT(SC/10):POKESP+7,Z+48:Z=SC-10*Z:POKESP+8,Z+48
1100 RETURN
1110 PRINT#15,"U1:"2;0;TR;SC:PRINT#15,"B-P:"2;0
1120 POKEBP,TR:POKEBP+1,SC:BP=BP+2:HI=INT(BP/256):LO=BP-(HI*256):POKEY1+11,LO
1130 POKEY1+12,HI:GOSUB1080:SYS(Y1):INPUT#15,EN$,ED$,ET$,ES$
1140 IFEN$<>"00"THEN1290
1150 BP=BP+256:IFBP+256<EBTHENRETURN
1160 POKE174,LO:POKE175,HI+1
1170 CLOSE2:CLOSE15
1180 GOSUB1200:GOSUB1070
1190 OPEN15,DI,15:OPEN2,DI,2,"#":RETURN
1200 IFBC=0THENOP$="PREMI RECORD & PLAY SUL REGISTR.":GOSUB1030:BC=1
1210 POKEY2+1,FC+48:FC=FC+1
1220 HI=INT(Y2/256):LO=Y2-(HI*256):POKE187,LO:POKE188,HI
1230 HI=INT(SB/256):LO=SB-(HI*256):POKE172,LO:POKE173,HI
1240 SYS(Y3):OP$="":GOSUB1030:RETURN
1250 OP$=BC$:GOSUB1030:FORA=1TO75:GETA$
1260 IFA$=""THENNEXT:OP$="":GOSUB1030:FORA=1TO75:GETA$:IFA$=""THENNEXT:GOTO1250
1270 OP$="":GOSUB1030:IFA$<>"█"THENRETURN
1280 GOTO60
1290 GOSUB1040:OP$="COPIA INCOMPLETA - RICOMINCIA DA CAPO":GOSUB1000
1300 GOSUB1000:BC$="ERRORE SUL DISCO ILLEGGIBILE":GOSUB1250:GOTO60
```

```
100 REM ***********************************
103 REM              LISTATO N 19
105 REM ***********************************
110 REM                ISTRUZIONI
120 REM          SALVARE IL PROGRAMMA
130 REM          DARE 'POKE 44,24'
140 REM           'POKE 6144,0'
150 REM    RICARICARE IL PROGR. E DARE RUN
160 REM ***********************************
260 CLR:POKE53280,8:POKE53281,8:PRINTCHR$(147)CHR$(144)
270 PRINT"ATTENDERE QUALCHE ISTANTE, PREGO..."
280 FORA1=49152TO49187:READB1:C1=C1+B1:POKEA1,B1:NEXT
290 IFC1=4646THEN320:REM CHECKSUM DI ROUTINE SAVE DATA
300 PRINTCHR$(147)"ERRORE NELLE ISTRUZIONI DATA."
310 PRINT"VERIFICARE NUMERI DI LINEA 680-790":END
320 FORA2=2049TO4657:READB2:C2=C2+B2:POKEA2,B2:NEXT
330 IFC2=229312THEN360:REM CHECKSUM PER I DATA DEL PROGRAMMA
340 PRINTCHR$(147)"ERRORE NELLE ISTRUZIONI DATA."
350 PRINT"VERIFICARE NUMERI DI LINEA 1000-****":END
360 PRINTCHR$(147)"I DATI SONO STATI CARICATI."
370 PRINT"POSSIAMO PROCEDERE ALLA REGISTRAZIONE"
380 PRINT"DEL PROGRAMMA FINALE."
390 INPUT"SALVO SU NASTRO O DISCO (N/D)";DV$
400 IFDV$="N"THENK1=1:GOTO430
410 IFDV$="D"THENK1=8:GOTO430
420 GOTO390
430 PRINT"INSERISCI IL SUPPORTO MAGNETICO"
440 PRINT"NELL'APPOSITO DISPOSITIVO DI SCRITTURA."
450 PRINT"PREMI LO SPAZIO QUANDO VUOI PARTIRE."
460 GETA$:IFA$=""THEN460
470 IFA$=" "THEN490
480 GOTO460
490 NM$="25-COPYFILE 1.1":PRINTCHR$(17)
500 IF(K1=8)AND(NM$="")THENNM$="-EDISOFT-"
510 K2=LEN(NM$):IFK2>16THENK2=16
520 FORX=0TOK2-1:K$=MID$(NM$+CHR$(0),X+1,1)
530 K3=ASC(K$)
540 POKE49188+X,K3:REM SCRIVE NOME DEL FILE
550 NEXT X
560 POKE49155,K1:REM NUMERO PERIFERICA
570 POKE49162,K2:REM LUNGH.NOME FILE
580 POKE49171,01:REM INIZIO BYTE BASSO
590 POKE49175,08:REM INIZIO BYTE ALTO
600 POKE49181,16:REM FINE BYTE BASSO
610 POKE49183,11:REM FINE BYTE ALTO
620 SYS49152
630 FORZ=0TO2000:NEXT
640 POKE53280,0:POKE53281,0:PRINTCHR$(147)CHR$(129)
650 PRINTCHR$(17)CHR$(17)"IL PROGRAMMA E' STATO CORRETTAMENTE"
660 PRINT"REGISTRATO COL NOME "CHR$(30)LEFT$(NM$,16)
670 FORZ=0TO5000:NEXT:SYS64738
```

```
680 DATA 169,   1, 162
690 DATA   8, 160,   1
700 DATA  32, 186, 255
710 DATA 169,   5, 162
720 DATA  36, 160, 192
730 DATA  32, 189, 255
740 DATA 169,   1, 133
750 DATA 251, 169,   8
760 DATA 133, 252, 169
770 DATA 251, 162,   4
780 DATA 160,  11,  32
790 DATA 216, 255,  96
830 REM ******************
850 REM DATA DEL PROGRAMMA
870 REM ******************
1000 DATA  11,   8, 193,   7, 158
1010 DATA  50,  48,  54,  52,   0
1020 DATA   0,   0,   0,   0,   0
1030 DATA 169,   1, 141,  32, 208
1040 DATA 141,  33, 208, 169,   0
1050 DATA 160,  16,  32,  30, 171
1060 DATA  32,   7,  15, 201,  49
1070 DATA 144, 249, 201,  52, 176
1080 DATA 245,  32, 210, 255, 201
1090 DATA  51, 208,   3,  76, 152
1100 DATA   8, 201,  49, 208,   3
1110 DATA  76,  67,  12,  32,  68
1120 DATA 229, 169,   8, 170, 160
1130 DATA   0,  32, 186, 255, 169
1140 DATA   1, 162, 248, 160,  16
1150 DATA  32, 189, 255,  32, 192
1160 DATA 255, 169,   8,  32, 180
1170 DATA 255, 169,   0,  32, 150
1180 DATA 255,  32, 165, 255,  32
1190 DATA 165, 255,  76, 128,   8
1200 DATA  32, 165, 255, 133,  34
1210 DATA  32, 165, 255, 166,  34
1220 DATA  32, 205, 189,  32,  33
1230 DATA  15,  32, 165, 255, 240
1240 DATA   6,  32, 210, 255,  24
1250 DATA 144, 245,  32,  36,  15
1260 DATA  32, 165, 255,  32, 165
1270 DATA 255, 165, 144, 240, 216
1280 DATA 169,   8,  32, 195, 255
1290 DATA  32,   7,  15,  76,  16
1300 DATA   8,  32,  68, 229,  32
1310 DATA   0,  15,  32,   7,  15
1320 DATA  32,  36,  15,  32,  36
1330 DATA  15,  32,  13,  15, 169
1340 DATA   8, 170, 160,   0,  32
1350 DATA 186, 255, 169,   1, 162
1360 DATA 248, 160,  16,  32, 189
1370 DATA 255,  32, 192, 255, 169
1380 DATA   8,  32, 180, 255, 169
1390 DATA   0,  32, 150, 255, 160
1400 DATA   4,  32, 165, 255, 136
1410 DATA 208, 250,  32, 165, 255
1420 DATA 133,  34,  32, 165, 255
1430 DATA 166,  34,  32, 205, 189
1440 DATA  32,  33,  15,  32, 165
1450 DATA 255, 240,   6,  32, 210
1460 DATA 255,  24, 144, 245,  32
1470 DATA  36,  15,  32,  36,  15
1480 DATA  32, 165, 255,  32, 165
1490 DATA 255, 160,   0, 132, 251
1500 DATA  32, 165, 255, 133, 252
1510 DATA  32, 165, 255, 133, 253
1520 DATA 166, 252,  32, 205, 189
1530 DATA  32,  33,  15, 160,   0
1540 DATA  32, 165, 255,  72,  32
1550 DATA  41,  15, 104, 153,  64
1560 DATA   3, 240,   3, 200, 208
1570 DATA 240,  32, 165, 255,  32
1580 DATA 165, 255, 165, 144, 240
1590 DATA   3,  76, 208,   9, 165
1600 DATA 253, 208, 253, 165, 252
1610 DATA 201, 235, 144,  10, 169
1620 DATA  65, 160,  17,  32,  30
1630 DATA 171,  76, 202,   9, 165
1640 DATA 251, 201,  32, 144,  10
1650 DATA 169, 126, 160,  17,  32
1660 DATA  30, 171,  24, 144, 237
1670 DATA 169,   0, 133,   8, 169
1680 DATA  32, 133, 211, 169,  29
1690 DATA 160,  17,  32,  30, 171
1700 DATA  32,  79,  15, 201,  78
1710 DATA 240,  93, 201,  89, 208
1720 DATA 245, 169,  43, 160,  17
1730 DATA  32,  30, 171, 165, 251
1740 DATA  32, 176,  12, 234, 234
1750 DATA 162,   0, 232, 189,  64
1760 DATA   3, 201,  34, 208, 248
1770 DATA 134, 253, 232, 189,  64
1780 DATA   3, 201,  34, 240,   7
1790 DATA 145,  65, 234, 232, 200
1800 DATA 208, 242, 138, 164, 251
1810 DATA  24, 229, 253, 153,  32
1820 DATA  18, 165, 252, 153, 192
1830 DATA  18, 189,  64,   3, 208
1840 DATA  10, 169, 100, 160,  17
1850 DATA  32,  30, 171,  76, 202
1860 DATA   9, 201,  83, 240,   7
1870 DATA 201,  80, 240,   3, 232
1880 DATA 208, 230, 153,  64,  18
1890 DATA 230, 251,  24, 144,   7
1900 DATA 169,  54, 160,  17,  32
1910 DATA  30, 171,  32,  36,  15
1920 DATA  76, 251,   8, 169,   8
1930 DATA  32, 195, 255, 165, 251
1940 DATA 208,   3,  76,  16,   8
```

```
1950 DATA 169, 157, 160,  17,  32
1960 DATA  30, 171,  32,  79,  15
1970 DATA 201,  49, 240,   7, 201
1980 DATA  50, 208, 245, 169,   0
1990 DATA  44, 169, 255, 133, 254
2000 DATA 169,  15,  32, 195, 255
2010 DATA 169,   0, 160,  14,  32
2020 DATA  30, 171,  32,  79,  15
2030 DATA 201,  51, 240,  47, 201
2040 DATA  50, 240,  20, 201,  49
2050 DATA 208, 241, 169,  96, 141
2060 DATA 149,   8,  32,  59,   8
2070 DATA 169,  76, 141, 149,   8
2080 DATA  76, 245,   9, 169,  86
2090 DATA 141, 246,  16, 169, 147
2100 DATA 160,  14,  32,  30, 171
2110 DATA  32,  13,  15, 169,  73
2120 DATA 141, 246,  16,  76, 245
2130 DATA   9, 162,   0, 134, 252
2140 DATA 169,  22, 157,  96,  18
2150 DATA 169,   0, 133, 253, 138
2160 DATA 240,  10, 169, 250, 160
2170 DATA  16,  32,  30, 171,  32
2180 DATA  79,  15, 169, 175, 160
2190 DATA  14,  32,  30, 171, 164
2200 DATA 252, 190,  32,  18, 165
2210 DATA 252,  32, 176,  12, 234
2220 DATA 177,  65, 234,  32, 210
2230 DATA 255, 200, 202, 208, 246
2240 DATA 169,   2, 162,   8, 168
2250 DATA  32, 186, 255, 166, 252
2260 DATA 189,  32,  18, 133,  34
2270 DATA 165, 252,  32, 176,  12
2280 DATA 234, 162,   0, 177,  65
2290 DATA 234, 157,  64,   3, 200
2300 DATA 232, 198,  34, 208, 244
2310 DATA 160,   0, 185, 186,  14
2320 DATA 157,  64,   3, 200, 232
2330 DATA 192,   4, 144, 244, 164
2340 DATA 252, 185,  64,  18, 157
2350 DATA  61,   3, 138, 162,  64
2360 DATA 160,   3,  32, 189, 255
2370 DATA  32, 192, 255, 166, 252
2380 DATA 189,  96,  18, 160,   0
2390 DATA 132,  34, 133,  35, 173
2400 DATA  17, 208,  41, 239, 141
2410 DATA  17, 208, 169,  15, 162
2420 DATA   8, 168,  32, 186, 255
2430 DATA 169,   3, 162, 191, 160
2440 DATA  14,  32, 189, 255,  32
2450 DATA 192, 255, 162,   2,  32
2460 DATA  15, 243,  32,  31, 243
2470 DATA 165, 186,  32, 180, 255
2480 DATA 165, 185,  32, 150, 255
2490 DATA 160,   0,  32, 165, 255
2500 DATA  32,   8,  13, 166, 144
2510 DATA 240, 246,  32,  89,  15
2520 DATA   8, 169,   2,  32, 195
2530 DATA 255, 173,  17, 208,   9
2540 DATA  16, 141,  17, 208,  40
2550 DATA 144,   3,  32,  96,  13
2560 DATA 166, 252, 165,  34, 157
2570 DATA 160,  18, 165,  35, 157
2580 DATA 128,  18, 232, 228, 251
2590 DATA 176,  38,  24, 125, 192
2600 DATA  18, 176,  32, 201, 234
2610 DATA 176,  28, 230, 252, 230
2620 DATA 253, 165,  35,  24, 105
2630 DATA   1, 157,  96,  18, 173
2640 DATA  17, 208,  41, 239, 141
2650 DATA  17, 208, 169,  15,  32
2660 DATA 195, 255,  76,  81,  10
2670 DATA 165, 252,  56, 229, 253
2680 DATA 133, 252, 169, 106, 160
2690 DATA  14,  32,  30, 171,  32
2700 DATA  79,  15, 169, 164, 160
2710 DATA  14,  32,  30, 171, 164
2720 DATA 252, 190,  32,  18, 165
2730 DATA 252,  32, 176,  12, 234
2740 DATA 177,  65, 234,  32, 210
2750 DATA 255, 200, 202, 208, 246
2760 DATA 169,   2, 162,   8, 168
2770 DATA  32, 186, 255, 166, 252
2780 DATA 189,  32,  18, 133,  34
2790 DATA 165, 252,  32, 176,  12
2800 DATA 234, 162,   0, 177,  65
2810 DATA 234, 157,  64,   3, 232
2820 DATA 200, 198,  34, 208, 244
2830 DATA 160,   0, 185, 195,  14
2840 DATA 157,  64,   3, 200, 232
2850 DATA 192,   4, 144, 244, 164
2860 DATA 252, 185,  64,  18, 157
2870 DATA  61,   3, 138, 162,  64
2880 DATA 160,   3,  32, 189, 255
2890 DATA 173,  17, 208,  41, 239
2900 DATA 141,  17, 208,  32, 192
2910 DATA 255, 166, 252, 160,   0
2920 DATA 189,  96,  18, 132,  34
2930 DATA 133,  35, 189, 160,  18
2940 DATA 133,  36, 189, 128,  18
2950 DATA 133,  37, 165, 186,  32
2960 DATA 177, 255, 165, 185,  32
2970 DATA 147, 255,  32,  25,  13
2980 DATA  32, 168, 255, 165,  36
2990 DATA 197,  34, 165,  37, 229
3000 DATA  35, 176, 240, 169,   2
3010 DATA  32, 195, 255,  32,  89
3020 DATA  15, 144,   3,  32,  96
3030 DATA  13,  36, 254,  48,   3
3040 DATA  76,  44,  12, 198, 253
```

```
3050 DATA  48,   5, 230, 252,  76
3060 DATA  85,  11,  32, 160,  13
3070 DATA 230, 252, 173,  17, 208
3080 DATA   9,  16, 141,  17, 208
3090 DATA 166, 252, 228, 251, 176
3100 DATA   3,  76, 152,  12, 169
3110 DATA 204, 160,  14,  32,  30
3120 DATA 171,  32,  79,  15,  76
3130 DATA  16,   8, 198, 253,  16
3140 DATA   3,  76,   9,  12, 230
3150 DATA 252, 173,  17, 208,   9
3160 DATA  16, 141,  17, 208,  32
3170 DATA 126,  13,  76,  75,  11
3180 DATA 169, 227, 160,  13,  32
3190 DATA  30, 171,  32,  96, 165
3200 DATA 169,  78, 141, 254,   1
3210 DATA 169,  58, 141, 255,   1
3220 DATA 169,  15, 162,   8, 168
3230 DATA  32, 186, 255, 162,   2
3240 DATA 189, 254,   1, 240,   3
3250 DATA 232, 208, 248, 138, 162
3260 DATA 254, 160,   1,  32, 189
3270 DATA 255,  32, 192, 255,  32
3280 DATA  89,  15,   8, 169,  15
3290 DATA  32, 195, 255,  40, 176
3300 DATA   3,  76,  16,   8, 169
3310 DATA 213, 160,  13,  32,  30
3320 DATA 171,  32,  79,  15, 201
3330 DATA  89, 240, 188, 201,  78
3340 DATA 208, 245,  76,  16,   8
3350 DATA  32,  36,  15,  76,  59
3360 DATA  10,   0,   0,  32, 160
3370 DATA  13, 173,  17, 208,   9
3380 DATA  16, 141,  17, 208,  76
3390 DATA   0,  15,   0,   0, 160
3400 DATA   0,  10,  10, 132,  66
3410 DATA  10,  38,  66,  10,  38
3420 DATA  66, 133,  65, 165,  66
3430 DATA  24, 105,  19, 133,  66
3440 DATA 160,   0,  96,   0,   0
3450 DATA   0,   0,   0,   0,   0
3460 DATA   0,   0,   0,   0,   0
3470 DATA   0,   0,   0,   0,   0
3480 DATA   0,   0,   0,   0,   0
3490 DATA   0,   0,   0,   0,   0
3500 DATA   0,   0,   0,   0,   0
3510 DATA   0,   0,   0,   0,   0
3520 DATA   0,   0,   0,   0,   0
3530 DATA   0,   0,   0,   0,   0
3540 DATA   0,   0,   0,   0,   0
3550 DATA   0,   0,   0,   0, 120
3560 DATA 160,  52, 132,   1, 160
3570 DATA   0,  96,  32,   0,  13
3580 DATA 145,  34, 230,  34, 208
3590 DATA   2, 230,  35, 160,  55
3600 DATA 132,   1,  88,  96,  32
3610 DATA   0,  13, 177,  34,  76
3620 DATA  13,  13,   0,   0,   0
3630 DATA   0,   0,   0,   0,   0
3640 DATA   0,   0,   0,   0,   0
3650 DATA   0,   0,   0,   0,   0
3660 DATA   0,   0,   0,   0,   0
3670 DATA   0,   0,   0,   0,  32
3680 DATA 165, 255,  41,  15,  10
3690 DATA  10,  10,  10, 133,  87
3700 DATA  32, 165, 255,  41,  15
3710 DATA   5,  87,  96,  72,  74
3720 DATA  74,  74,  74,  32,  88
3730 DATA  13, 104,  41,  15,  24
3740 DATA 105,  48,  76, 210, 255
3750 DATA 173,  17, 208,   9,  16
3760 DATA 141,  17, 208, 169, 224
3770 DATA 160,  14,  32,  30, 171
3780 DATA  32,  79,  15, 201,  89
3790 DATA 208,   1,  96, 201,  78
3800 DATA 208, 244,  76,  62,  15
3810 DATA 169, 255, 160,  17,  32
3820 DATA  30, 171, 165, 252,  10
3830 DATA  10,  10,  10, 166, 252
3840 DATA 188,  32,  18, 170, 189
3850 DATA   0,  19,  32, 210, 255
3860 DATA 232, 136, 208, 246,  96
3870 DATA   0,   0,   0,   0, 173
3880 DATA  17, 208,  41, 239, 141
3890 DATA  17, 208, 169,  15,  32
3900 DATA 195, 255, 169,  15, 162
3910 DATA   8, 168,  32, 186, 255
3920 DATA 169,   3, 162, 200, 160
3930 DATA  14,  32, 189, 255,  32
3940 DATA 192, 255, 169,  15,  76
3950 DATA 195, 255,   0,   0,  32
3960 DATA 160,  13, 173,  17, 208
3970 DATA   0,   0,   0,   0,   0
3980 DATA   0,   0,  13,  13,  65
3990 DATA  78,  67,  79,  82,  65
4000 DATA  32,  32,  32,  32,  63
4010 DATA   0,  13,  13,  68,  73
4020 DATA  77,  77,  73,  32,  78
4030 DATA  79,  77,  69,  32,  69
4040 DATA  32,  73,  68,  32,  68
4050 DATA  69,  76,  32,  68,  75
4060 DATA  32,  58,  13,  13,   0
4070 DATA  13,  13,  67,  72,  69
4080 DATA  32,  86,  85,  79,  73
4090 DATA  32,  70,  65,  82,  69
4100 DATA  32,  63,  32,  46,  46
4110 DATA  46,  13,  13,  18,  32
4120 DATA  49,  32, 146,  32,  46
4130 DATA  46,  46,  32,  86,  69
4140 DATA  68,  69,  82,  69,  32
```

```
4150 DATA  76,  65,  32,  68,  73
4160 DATA  82,  69,  67,  84,  79
4170 DATA  82,  89,  13,  13,  18
4180 DATA  32,  50,  32, 146,  32
4190 DATA  46,  46,  46,  32,  86
4200 DATA  65,  76,  73,  68,  65
4210 DATA  84,  69,  32,  68,  69
4220 DATA  76,  32,  68,  75,  13
4230 DATA  13,  18,  32,  51,  32
4240 DATA 146,  32,  46,  46,  46
4250 DATA  32,  73,  78,  73,  90
4260 DATA  73,  79,  32,  67,  79
4270 DATA  80,  73,  65,  32,  13
4280 DATA   0,  13,  13,  77,  69
4290 DATA  84,  84,  73,  32,  32
4300 DATA  28,  18,  73,  76,  32
4310 DATA  68,  75,  32,  68,  69
4320 DATA  83,  84,  73, 146,  31
4330 DATA  32,  78,  65,  90,  73
4340 DATA  79,  78,  69,  32,  78
4350 DATA  69,  76,  32,  68,  82
4360 DATA  45,   0,  13,  13,  86
4370 DATA  65,  76,  73,  68,  65
4380 DATA  84,  73,  78,  71,  32
4390 DATA  46,  46,  46,   0,  13
4400 DATA  13,  83,  67,  82,  73
4410 DATA  86,  79,  32,  32,   0
4420 DATA  13,  13,  32,  76,  69
4430 DATA  71,  71,  79,  32,  32
4440 DATA   0,  44,  88,  44,  82
4450 DATA   0,  85,  73,  45,   0
4460 DATA  44,  88,  44,  87,   0
4470 DATA  85,  73,  43,   0,  13
4480 DATA  13,  18,  28,  70,  73
4490 DATA  78,  69,  32,  67,  79
4500 DATA  80,  73,  65,  32,  32
4510 DATA  33, 146,  31,   0,  13
4520 DATA  13,  86,  85,  79,  73
4530 DATA  32,  67,  79,  78,  84
4540 DATA  73,  78,  85,  65,  82
4550 DATA  69,  32,  65,  32,  67
4560 DATA  79,  80,  73,  65,  82
4570 DATA  69,  63,   0,   0,   0
4580 DATA   0, 169, 250, 160,  16
4590 DATA  76,  30, 171,  32, 228
4600 DATA 255, 240, 251,  96, 169
4610 DATA  15, 162,   8, 168,  32
4620 DATA 186, 255, 169,   1, 162
4630 DATA 246, 160,  16,  32, 189
4640 DATA 255,  76, 192, 255, 169
4650 DATA  32,  44, 169,  13,  76
4660 DATA 210, 255,  32, 210, 255
4670 DATA 138,  72, 152,  72,  32
4680 DATA 225, 255,  24, 208,   1
4690 DATA  56, 104, 168, 104, 170
4700 DATA 176,   1,  96, 162, 246
4710 DATA 154,  32, 160,  12, 169
4720 DATA  15,  32, 195, 255,  32
4730 DATA 231, 255,  76,  16,   8
4740 DATA  32, 228, 255, 208, 233
4750 DATA  32,  44,  15, 144, 246
4760 DATA 162,  15,  32,  15, 243
4770 DATA  32,  31, 243, 165, 186
4780 DATA  32, 180, 255, 165, 185
4790 DATA  32, 150, 255,  32,  60
4800 DATA  13, 201,  32,   8, 144
4810 DATA  11,  72,  32,  36,  15
4820 DATA  32,  36,  15, 104,  32
4830 DATA  79,  13,  32, 165, 255
4840 DATA 201,  13, 240,   9,  40
4850 DATA   8, 144, 245,  32, 210
4860 DATA 255, 144, 240,  32, 171
4870 DATA 255,  40,  96,   0,   0
4880 DATA   0,   0,   0,   0,   0
4890 DATA   0,   0,   0,   0,   0
4900 DATA   0,   0,   0,   0,   0
4910 DATA   0,   0,   0,   0,   0
4920 DATA   0,   0,   0,   0,   0
4930 DATA   0,   0,   0,   0,   0
4940 DATA   0,   0,   0,   0,   0
4950 DATA   0,   0,   0,   0,   0
4960 DATA   0,   0,   0,   0,   0
4970 DATA   0,   0,   0,   0,   0
4980 DATA   0,   0,   0,  52,  56
4990 DATA  57,  32,  32,  52,  57
5000 DATA  57,  48,  32,  52,  57
5010 DATA  56,  48,  32,  52,  57
5020 DATA  55,  48,  32,  52,  57
5030 DATA  54,  48,  32,  52,  57
5040 DATA  52,  32,  32,  53,  48
5050 DATA  52,  48,  32,  53,  48
5060 DATA  51,  48,  32,  53,  48
5070 DATA  50,  48,  32,  53,  48
5080 DATA  49,  48,  32,  53,  48
5090 DATA  48,  48, 147,  31,  32
5100 DATA  32,  32,  32,  32,  67
5110 DATA  79,  80,  73,  65,  32
5120 DATA  68,  73,  32,  70,  73
5130 DATA  76,  69,  32,  80,  69
5140 DATA  82,  32,  67,  54,  52
5150 DATA  32,  38,  32,  49,  53
5160 DATA  52,  49,  13,  32,  32
5170 DATA  32,  32,  32,  45,  45
5180 DATA  45,  45,  45,  45,  45
5190 DATA  45,  45,  45,  45,  45
5200 DATA  45,  45,  45,  45,  45
5210 DATA  45,  45,  45,  45,  45
5220 DATA  45,  45,  45,  45,  45
5230 DATA  45,  13,  13,  32,  32
5240 DATA  32,  32,  32,  18,  40
```

```
5250 DATA  67,  41, 146,  32,  32
5260 DATA  49,  57,  56,  51,  32
5270 DATA  66,  89,  32,  67,  46
5280 DATA  32,  67,  69,  82,  82
5290 DATA  79,  78,  73,  13,  13
5300 DATA  32,  32,  32,  32,  32
5310 DATA  80,  85,  66,  66,  76
5320 DATA  73,  67,  79,  32,  68
5330 DATA  79,  77,  73,  78,  73
5340 DATA  79,  32,  49,  57,  56
5350 DATA  53,  32,  42,  13,  13
5360 DATA  32,  32,  32,  32,  32
5370 DATA  32,  32,  32,  32,  32
5380 DATA  32,  32,  42,  42,  42
5390 DATA  42,  42,  42,  42,  13
5400 DATA  13,  13,  13,  18,  32
5410 DATA  49,  32, 146,  32,  70
5420 DATA  79,  82,  77,  65,  84
5430 DATA  84,  65,  90,  73,  79
5440 DATA  78,  69,  32,  32,  13
5450 DATA  13,  18,  32,  50,  32
5460 DATA 146,  32,  67,  65,  82
5470 DATA  73,  67,  65,  32,  32
5480 DATA  68,  73,  82,  69,  67
5490 DATA  84,  79,  82,  89,  13
5500 DATA  13,  18,  32,  51,  32
5510 DATA 146,  32,  67,  79,  80
5520 DATA  73,  65,  32,  70,  73
5530 DATA  76,  69,  13,  13,  13
5540 DATA  68,  73,  71,  73,  84
5550 DATA  65,  32,  76,  65,  32
5560 DATA  83,  67,  69,  76,  84
5570 DATA  65,  32,  32,  32,  18
5580 DATA  63, 157,   0,  73,   0
5590 DATA  36,   0,  13,  77,  69
5600 DATA  84,  84,  73,  32,  73
5610 DATA  76,  32,  68,  73,  83
5620 DATA  67,  79,  32,  28,  18
5630 DATA  83,  79,  82,  71,  46
5640 DATA 146,  31,  32,  78,  69
5650 DATA  76,  32,  68,  82,  46
5660 DATA  45,   0,  28,  89,  69
5670 DATA  83,  47,  78,  79, 157
5680 DATA 157, 157, 157, 157, 157
5690 DATA   0,  18,  30,  89,  69
5700 DATA  83, 146,  31,  32,  32
5710 DATA  32,   0,  30,  18,  78
5720 DATA  79, 146,  31,  32,  32
5730 DATA  32,  32,   0,  13,  32
5740 DATA  28,  18,  94, 146,  31
5750 DATA  32,  70,  73,  76,  69
5760 DATA  32,  84,  82,  79,  80
5770 DATA  80,  79,  32,  76,  85
5780 DATA  78,  71,  79,  32,  68
5790 DATA  65,  32,  67,  79,  80
5800 DATA  46,  46,   0,  13,  32
5810 DATA  18,  28,  94, 146,  31
5820 DATA  32,  70,  73,  76,  69
5830 DATA  32,  73,  76,  76,  69
5840 DATA  71,  65,  76,  69,  32
5850 DATA  32,  33,  32,   0,  13
5860 DATA  32,  76,  65,  32,  76
5870 DATA  73,  83,  84,  65,  32
5880 DATA  68,  65,  32,  67,  79
5890 DATA  80,  46,  32,  69,  39
5900 DATA  18,  28,  80,  73,  69
5910 DATA  78,  65, 146,  31,   0
5920 DATA  13,  13,  86,  85,  79
5930 DATA  73,  32,  67,  79,  80
5940 DATA  73,  65,  82,  69,  32
5950 DATA  73,  32,  80,  82,  79
5960 DATA  71,  82,  65,  77,  77
5970 DATA  73,  32,  73,  78,  32
5980 DATA  77,  69,  77,  46,  32
5990 DATA  46,  46,  46,  13,  13
6000 DATA  18,  32,  49,  32, 146
6010 DATA  32,  46,  46,  46,  32
6020 DATA  83,  69,  78,  90,  65
6030 DATA  32,  83,  79,  83,  84
6040 DATA  65,  13,  13,  18,  32
6050 DATA  50,  32, 146,  32,  46
6060 DATA  46,  46,  32,  79,  32
6070 DATA  79,  71,  78,  73,  32
6080 DATA  70,  73,  76,  69,  32
6090 DATA  83,  69,  80,  65,  82
6100 DATA  65,  84,  79,  32,  32
6110 DATA  63,  13,   0,  13,  13
6120 DATA  80,  82,  88,  46,  32
6130 DATA  70,  73,  76,  69,  32
6140 DATA  58,  32,   0,   0,   0
6150 DATA 169,  99,  85, 112, 255
6160 DATA  32,  76, 255,   0,   8
6170 DATA   0,  49,  32,   0, 181
6180 DATA  40, 160,   0,   0,   7
6190 DATA   0,   0, 255,   0,  79
6200 DATA 160,   0,   0,   0,   0
6210 DATA   0,   0,   0,   0
```

```
100 REM ****************************
103 REM      LISTATO N 20
105 REM ****************************
110 REM       ISTRUZIONI
120 REM  TERMINATA LA BATTITURA,
130 REM  SALVARE IL PROGRAMMA.
140 REM  DARE POKE44,24:POKE6144,0
150 REM  RICARICARE IL PRG. E RUN
160 REM ****************************
260 CLR:POKE53280,8:POKE53281,8:PRINTCHR$(147)CHR$(144)
270 PRINT"██████████████ATTENDERE QUALCHE ISTANTE, PREGO..."
280 FORA1=49152TO49187:READB1:C1=C1+B1:POKEA1,B1:NEXT
290 IFC1=4646THEN320:REM CHECKSUM DI ROUTINE SAVE DATA
300 PRINTCHR$(147)"██ERRORE NELLE ISTRUZIONI DATA."
310 PRINT"VERIFICARE NUMERI DI LINEA 680-790":END
320 FORA2=2049TO3272:READB2:C2=C2+B2:POKEA2,B2:NEXT
330 IFC2=113601THEN360:REM CHECKSUM PER I DATA DEL PROGRAMMA
340 PRINTCHR$(147)"██ERRORE NELLE ISTRUZIONI DATA."
350 PRINT"VERIFICARE NUMERI DI LINEA 1000-****":END
360 PRINTCHR$(147)"I DATI SONO STATI CARICATI."
370 PRINT"POSSIAMO PROCEDERE ALLA REGISTRAZIONE"
380 PRINT"DEL PROGRAMMA FINALE.██"
390 INPUT"SALVO SU NASTRO O DISCO (N/D)";DV$
400 IFDV$="N"THENK1=1:GOTO430
410 IFDV$="D"THENK1=8:GOTO430
420 GOTO390
430 PRINT"██INSERISCI IL SUPPORTO MAGNETICO"
440 PRINT"NELL'APPOSITO DISPOSITIVO DI SCRITTURA."
450 PRINT"██PREMI LO SPAZIO QUANDO VUOI PARTIRE."
460 GETA$:IFA$=""THEN460
470 IFA$=" "THEN490
480 GOTO460
490 NM$="34-COPIA INTERO":PRINTCHR$(17)
500 IF(K1=8)AND(NM$="")THENNM$="-EDISOFT-"
510 K2=LEN(NM$):IFK2>16THENK2=16
520 FORX=0TOK2-1:K$=MID$(NM$+CHR$(0),X+1,1)
530 K3=ASC(K$)
540 POKE49188+X,K3:REM SCRIVE NOME DEL FILE
550 NEXT X
560 POKE49155,K1:REM NUMERO PERIFERICA
570 POKE49162,K2:REM LUNGH.NOME FILE
580 POKE49171,01:REM INIZIO BYTE BASSO
590 POKE49175,08:REM INIZIO BYTE ALTO
600 POKE49181,16:REM FINE BYTE BASSO
610 POKE49183,11:REM FINE BYTE ALTO
620 SYS49152
630 FORZ=0TO2000:NEXT
640 POKE53280,0:POKE53281,0:PRINTCHR$(147)CHR$(129)
650 PRINTCHR$(17)CHR$(17)"IL PROGRAMMA E' STATO CORRETTAMENTE"
660 PRINT"REGISTRATO COL NOME "CHR$(30)LEFT$(NM$,16)
670 FORZ=0TO5000:NEXT:SYS64738
```

```
680 DATA 169,   1, 162
690 DATA   8, 160,   1
700 DATA  32, 186, 255
710 DATA 169,   5, 162
720 DATA  36, 160, 192
730 DATA  32, 189, 255
740 DATA 169,   1, 133
750 DATA 251, 169,   8
760 DATA 133, 252, 169
770 DATA 251, 162,   4
780 DATA 160,  11,  32
790 DATA 216, 255,  96
830 REM ******************
850 REM DATA DEL PROGRAMMA
870 REM ******************
1000 DATA  11,   8,   0,   0, 158
1010 DATA  50,  48,  54,  49,   0
1020 DATA   0,   0, 169, 151, 141
1030 DATA   0, 221, 169, 125, 162
1040 DATA  11, 133,   3, 134,   4
1050 DATA 169,   0, 162,   5, 133
1060 DATA   5, 134,   6, 169,   8
1070 DATA  32,  12, 237, 169, 111
1080 DATA  32, 185, 237, 169,  77
1090 DATA  32, 221, 237, 169,  45
1100 DATA  32, 221, 237, 169,  87
1110 DATA  32, 221, 237, 160,   0
1120 DATA 165,   5,  32, 221, 237
1130 DATA 165,   6,  32, 221, 237
1140 DATA 169,  30,  32, 221, 237
1150 DATA 177,   3,  32, 221, 237
1160 DATA 200, 192,  30, 144, 246
1170 DATA  32, 254, 237,  24, 165
1180 DATA   3, 105,  30, 133,   3
1190 DATA 144,   3, 230,   4,  24
1200 DATA 165,   5, 166,   6, 105
1210 DATA  30, 133,   5, 144,   2
1220 DATA 230,   6, 224,   6, 144
1230 DATA 173, 201,  47, 144, 169
1240 DATA 169,   8,  32,  12, 237
1250 DATA 169, 111,  32, 185, 237
1260 DATA 169,  85,  32, 221, 237
1270 DATA 169,  67,  32, 221, 237
1280 DATA  32, 254, 237, 120, 162
1290 DATA   0, 202, 208, 253, 162
1300 DATA   0, 202, 208, 253,  32
1310 DATA  87,  11, 147,  17,  32
1320 DATA  32,  32,  67,  79,  80
1330 DATA  73,  65,  84,  79,  82
1340 DATA  69,  32,  51,  30,  48
1350 DATA  32,  45,  32,  80,  69
1360 DATA  82,  32,  67,  61,  54
1370 DATA  52,  32,  38,  32,  49
1380 DATA  53,  52,  49,  32,  13
1390 DATA  17,  32,  32,  32,  32
```

```
1400 DATA  32,  40,  67,  41,  32
1410 DATA  79,  84,  84,  30,  56
1420 DATA  51,  32,  67,  76,  65
1430 DATA  85,  68,  73,  79,  32
1440 DATA  67,  69,  82,  82,  79
1450 DATA  78,  73,  13,   0, 169
1460 DATA   1, 133,   5, 169,   0
1470 DATA 133,   6, 169,  54, 133
1480 DATA   1, 169,  27, 141,  17
1490 DATA 208,  32,  87,  11,  17
1500 DATA  18,  32,  68,  73,  83
1510 DATA  67,  79,  32,  79,  82
1520 DATA  73,  71,  73,  78,  65
1530 DATA  76,  69,  32,  32, 146
1540 DATA  17,  13,   0,  88,  32
1550 DATA 228, 255, 240, 251, 169
1560 DATA  11, 141,  17, 208, 162
1570 DATA  80, 160,   0, 136, 208
1580 DATA 253, 202, 208, 250, 120
1590 DATA 169,   0, 133,   3, 169
1600 DATA  24, 133,   4, 165,   6
1610 DATA 166,   5, 141,  51,  11
1620 DATA 142,  50,  11, 165,   5
1630 DATA 164,   6,  32,   9,  11
1640 DATA 165,   2, 201,   2, 144
1650 DATA   3,  32, 127,  10, 230
1660 DATA   6, 166,   5, 189,  51
1670 DATA  11, 197,   6, 176,  12
1680 DATA 169,   0, 133,   6, 230
1690 DATA   5, 165,   5, 201,  36
1700 DATA 176,   6, 230,   4, 165
1710 DATA   4, 208, 211, 169,  54
1720 DATA 133,   1, 169,  27, 141
1730 DATA  17, 208,  32,  87,  11
1740 DATA  17,  18,  32,  32,  32
1750 DATA  32,  68,  73,  83,  67
1760 DATA  79,  32,  67,  79,  80
1770 DATA  73,  65,  32,  32,  32
1780 DATA 146,  17,  13,   0,  88
1790 DATA  32, 228, 255, 240, 251
1800 DATA 169,  11, 141,  17, 208
1810 DATA 162,  80, 160,   0, 136
1820 DATA 208, 253, 202, 208, 250
1830 DATA 120, 169,   0, 133,   3
1840 DATA 169,  24, 133,   4, 173
1850 DATA  50,  11, 174,  51,  11
1860 DATA 133,   5, 134,   6, 165
1870 DATA   5, 164,   6,  32,   7
1880 DATA  11, 165,   2, 201,   2
1890 DATA 144,   3,  32,  88,  10
1900 DATA 230,   6, 166,   5, 189
1910 DATA  51,  11, 197,   6, 176
1920 DATA  12, 169,   0, 133,   6
1930 DATA 230,   5, 165,   5, 201
1940 DATA  36, 176,   9, 230,   4
```

```
1950 DATA 165,   4, 208, 211,  76
1960 DATA 238,   8, 169,   0,  32
1970 DATA 163,  10, 169, 151, 141
1980 DATA   0, 221, 169,  55, 133
1990 DATA   1, 169,  27, 141,  17
2000 DATA 208,  88,  32,  87,  11
2010 DATA  17,  17,  32,  32,  32
2020 DATA  32,  32,  32,  67,  79
2030 DATA  80,  73,  65,  32,  85
2040 DATA  76,  84,  73,  77,  65
2050 DATA  84,  65,  32,  48,  48
2060 DATA  32,  79,  75,  32,  17
2070 DATA  13,   0,  76, 123, 227
2080 DATA 165,   5,  32,  56,  10
2090 DATA  32,  87,  11,  32,  83
2100 DATA  69,  84,  84,  30,  32
2110 DATA  32,   0, 165,   6,  32
2120 DATA  56,  10, 169,  13,  76
2130 DATA 210, 255, 248, 170, 169
2140 DATA   0, 224,   0, 240,   7
2150 DATA  24, 105,   1, 202,  76
2160 DATA  60,  10, 216,  72,  74
2170 DATA  74,  74,  74,  32,  81
2180 DATA  10, 104,  41,  15,   9
2190 DATA  48,  76, 210, 255, 169
2200 DATA  54, 133,   1,  32,  87
2210 DATA  11,  69,  82,  82,  30
2220 DATA  32,  84,  82,  79,  86
2230 DATA  65,  84,  79,  32,  32
2240 DATA  32,  58,  32,  84,  82
2250 DATA  30,  32,  32,   0,  32
2260 DATA  29,  10, 120, 169,  52
2270 DATA 133,   1,  96, 169,  54
2280 DATA 133,   1,  32,  87,  11
2290 DATA  69,  82,  82,  30,  32
2300 DATA  84,  82,  79,  86,  65
2310 DATA  84,  79,  58,  32,  84
2320 DATA  82,  30,  32,  32,   0
2330 DATA  32,  29,  10, 120, 169
2340 DATA  52, 133,   1,  96, 133
2350 DATA 107, 169,  53, 133,   1
2360 DATA 169,  11, 141,   0, 221
2370 DATA 173,   0, 221,  16, 251
2380 DATA 169,   3, 141,   0, 221
2390 DATA 162,   4, 169,   3,  70
2400 DATA 107, 106,  70, 107, 106
2410 DATA  74,  74, 234, 141,   0
2420 DATA 221, 202, 208, 239, 162
2430 DATA   1, 202, 208, 253, 169
2440 DATA  52, 133,   1,  96, 169
2450 DATA  53, 133,   1, 169,  11
2460 DATA 141,   0, 221, 173,   0
2470 DATA 221,  16, 251, 169,   3
2480 DATA 141,   0, 221, 162,   5
2490 DATA 202, 234, 208, 252, 162
2500 DATA   4, 173,   0, 221,  10
2510 DATA   8,  10,  38, 107,  40
2520 DATA  38, 107, 202, 208, 242
2530 DATA 169,  52, 133,   1, 165
2540 DATA 107,  73, 255,  96,   9
2550 DATA 128,  72,  32, 163,  10
2560 DATA 152,  32, 163,  10, 104
2570 DATA  72,  16,  10, 160,   0
2580 DATA 177,   3,  32, 163,  10
2590 DATA 200, 208, 248,  32, 213
2600 DATA  10, 133,   2, 104,  48
2610 DATA  10, 160,   0,  32, 213
2620 DATA  10, 145,   3, 200, 208
2630 DATA 248,  96,   0,   0,  20
2640 DATA  20,  20,  20,  20,  20
2650 DATA  20,  20,  20,  20,  20
2660 DATA  20,  20,  20,  20,  20
2670 DATA  20,  18,  18,  18,  18
2680 DATA  18,  18,  18,  17,  17
2690 DATA  17,  17,  17,  17,  16
2700 DATA  16,  16,  16,  16, 104
2710 DATA 133,  53, 104, 133,  54
2720 DATA 160,   0, 230,  53, 208
2730 DATA   2, 230,  54, 177,  53
2740 DATA 240,  11,  32, 210, 255
2750 DATA 230,  53, 208, 245, 230
2760 DATA  54, 208, 241, 230,  53
2770 DATA 208,   2, 230,  54, 108
2780 DATA  53,   0, 120, 173,   0
2790 DATA  28,   9,  12, 141,   0
2800 DATA  28, 162,   0, 202, 208
2810 DATA 253, 173,   0,  24,   9
2820 DATA  10,  41, 239, 141,   0
2830 DATA  24,  32, 141,   5, 208
2840 DATA   3, 108, 252, 255, 133
2850 DATA 128,  32, 141,   5, 133
2860 DATA 129, 173,   0,  28,  73
2870 DATA   8, 141,   0,  28, 165
2880 DATA 128,  16,  14, 160,   0
2890 DATA  32, 141,   5, 153,   0
2900 DATA   3, 200, 208, 247, 169
2910 DATA 144,  44, 169, 128, 133
2920 DATA 130,  32, 189,   5,  32
2930 DATA  93,   5, 165, 128,  48
2940 DATA 201, 160,   0, 185,   0
2950 DATA   3,  32,  93,   5, 200
2960 DATA 208, 247,  76,  24,   5
2970 DATA 133, 130,  44,   0,  24
2980 DATA  16, 251, 169,  16, 141
2990 DATA   0,  24,  44,   0,  24
3000 DATA  48, 251, 162,   4, 169
3010 DATA   0,   6, 130,  42,  10
3020 DATA   6, 130,  42,  10, 141
3030 DATA   0,  24, 202, 208, 240
3040 DATA 162,   1, 202, 208, 254
```

```
3050 DATA 234, 234, 169,  15, 141
3060 DATA   0,  24,  96,  44,   0
3070 DATA  24,  16, 251, 169,  16
3080 DATA 141,   0,  24,  44,   0
3090 DATA  24,  48, 251, 162,   4
3100 DATA 202, 208, 253, 142,   0
3110 DATA  24, 162,   4, 173,   0
3120 DATA  24,  74,   8,  74,  74
3130 DATA 102, 130,  40, 102, 130
3140 DATA 202, 208, 241, 169,  15
3150 DATA 141,   0,  24, 165, 130
3160 DATA  96,  88, 165, 128,  41
3170 DATA 127, 133,   6, 165, 129
3180 DATA 133,   7, 169,   1, 133
3190 DATA  10, 165, 130, 133,   0
3200 DATA 165,   0,  16,  11, 201
3210 DATA 160, 208, 248, 169, 240
3220 DATA 133,   0,  76, 208,   5
3230 DATA 201,   3, 144,  72, 201
3240 DATA   8, 240,  68, 201,  15
3250 DATA 240,  64,  36,  10,  48
3260 DATA  60, 201,  11, 240,   7
3270 DATA 198,  10, 208,  13, 169
3280 DATA 192,  44, 169, 176, 133
3290 DATA   0, 165,   0,  48, 252
3300 DATA  16, 200, 169,   0, 133
3310 DATA  10, 166,  10, 189, 219
3320 DATA 254, 240,  28, 141, 254
3330 DATA   2, 173, 254,   2,  48
3340 DATA 251, 165, 130, 201, 144
3350 DATA 208,   2, 169, 160, 133
3360 DATA   0, 165,   0,  48, 252
3370 DATA 230,  10, 201,   1, 208
3380 DATA 221, 165,   0, 120,  96
3390 DATA   0,   0,   0,   0, 167
3400 DATA  70,  38, 255,   0,  25
3410 DATA 148,  56, 240, 105,   1
3420 DATA   0, 239, 207,   0,  40
3430 DATA 255,  29, 202,   0,   8
3440 DATA   0,   0,   0,   0,   0
```

```
100 REM ******************************
103 REM       LISTATO N 21
105 REM ******************************
110 REM        ISTRUZIONI
120 REM   SALVARE IL PROGRAMMA
130 REM   DARE'POKE 44,24:POKE6144,0
140 REM   RICARICARE DIGITANDO RUN
160 REM ******************************
260 CLR:POKE53280,8:POKE53281,8:PRINTCHR$(147)CHR$(144)
270 PRINT"ATTENDERE QUALCHE ISTANTE, PREGO..."
280 FORA1=49152TO49187:READB1:C1=C1+B1:POKEA1,B1:NEXT
290 IFC1=4646THEN320:REM CHECKSUM DI ROUTINE SAVE DATA
300 PRINTCHR$(147)"ERRORE NELLE ISTRUZIONI DATA."
310 PRINT"VERIFICARE NUMERI DI LINEA 680-790":END
320 FORA2=2049TO4585:READB2:C2=C2+B2:POKEA2,B2:NEXT:REM **
330 IFC2=235376THEN360:REM CHECKSUM PER I DATA DEL PROGRAMMA
340 PRINTCHR$(147)"ERRORE NELLE ISTRUZIONI DATA."
350 PRINT"VERIFICARE NUMERI DI LINEA 1000-****":END
360 PRINTCHR$(147)"I DATI SONO STATI CARICATI."
370 PRINT"POSSIAMO PROCEDERE ALLA REGISTRAZIONE"
380 PRINT"DEL PROGRAMMA FINALE."
390 INPUT"SALVO SU NASTRO O DISCO (N/D)";DV$
400 IFDV$="N"THENK1=1:GOTO430
410 IFDV$="D"THENK1=8:GOTO430
420 GOTO390
430 PRINT"INSERISCI IL SUPPORTO MAGNETICO"
440 PRINT"NELL'APPOSITO DISPOSITIVO DI SCRITTURA."
450 PRINT"PREMI LO SPAZIO QUANDO VUOI PARTIRE."
460 GETA$:IFA$=""THEN460
470 IFA$=" "THEN490
480 GOTO460
490 NM$="24-COPIAT.E.23":PRINTCHR$(17)
500 IF(K1=8)AND(NM$="")THENNM$="-EDISOFT-"
510 K2=LEN(NM$):IFK2>16THENK2=16
520 FORX=0TOK2-1:K$=MID$(NM$+CHR$(0),X+1,1)
530 K3=ASC(K$)
540 POKE49188+X,K3:REM SCRIVE NOME DEL FILE
550 NEXT X
560 POKE49155,K1:REM NUMERO PERIFERICA
570 POKE49162,K2:REM LUNGH.NOME FILE
580 POKE49171,01:REM INIZIO BYTE BASSO
590 POKE49175,08:REM INIZIO BYTE ALTO
600 POKE49181,16:REM FINE BYTE BASSO
610 POKE49183,11:REM FINE BYTE ALTO
620 SYS49152
630 FORZ=0TO2000:NEXT
640 POKE53280,0:POKE53281,0:PRINTCHR$(147)CHR$(129)
650 PRINTCHR$(17)CHR$(17)"IL PROGRAMMA E' STATO CORRETTAMENTE"
660 PRINT"REGISTRATO COL NOME "CHR$(30)LEFT$(NM$,16)
670 FORZ=0TO5000:NEXT:SYS64738
```